

## Treball final de grau

**Estudi:** Grau en Enginyeria Informàtica

**Títol:** Sistema recomanador d'events en una xarxa social

**Document:** Memòria

**Alumne:** Maria del Mar Llambí Morillas

**Director/tutor:** Esteve del Acebo

**Departament:** Informàtica, matemàtica aplicada i estadística

**Àrea:** Llenguatges i sistemes informàtics

**Convocatòria (mes/any):** Febrer 2016



“No hay enigmas. Si un problema puede plantearse, también puede resolverse.”

**Ludwig Wittgenstein** (Filòsof i analista)

## Agraïments

Aquest projecte no hauria estat possible sense el recolzament incondicional dels meus pares i dels meus amics. Al meu germà, per ensenyar-me a lluitar. Al meu tutor Esteve del Acebo, per ajudar-me a prendre les millors decisions. Als meus companys de pis, per aquelles tantíssimes hores tancada a l'habitació. A tot l'equip del Trueta, per ajudar-me a aguantar i aconseguir redactar tot el projecte a temps. A tu, per estar al meu costat.

## Índex

1. Introducció, motivacions, propòsit i objectius del projecte .....	8
1.1 - Motivacions .....	9
1.2- Propòsits i objectius .....	10
2. Estudi de viabilitat .....	11
3. Metodologia .....	12
4. Planificació.....	13
5. Marc de treball i conceptes prèvis .....	17
5.1 - Xarxa social.....	17
5.2 - Data analytics & predictive analytics.....	18
5.3 - Sistema Recomanador .....	19
5.3.1 - Sistema Col·laboratiu.....	21
5.3.2 - Nearest Neighborhood i All Neighborhood .....	23
5.3.3 - Filtratge basat en el contingut .....	24
5.3.4 - Filtratge basat en el coneixement .....	25
5.4 - Sistema de control de versions i Git .....	26
5.4.1 - Git i avantatges .....	28
5.5 - Model-Vista-Controlador.....	28
5.5.1- Els components del MVC .....	29
5.5.2 - Els avantatges del MVC .....	29
5.6 - L'entorn del projecte .....	30
5.6.1 - Startup .....	30
6. Requisits del sistema .....	31
6.1 - Requisits de l'aplicació web: Boonobo.....	31
6.1.1 - Requisits funcionals .....	31
6.1.2 - Requisits no funcionals .....	32
6.2 - Requisits del sistema recomanador.....	33
6.2.1 - Requeriments no funcionals.....	33
7. Estudis i decisions.....	33
7.1 - Llenguatges de programació .....	33
7.1.1 - PHP & HTML5/CSS .....	33
7.1.2 - Java .....	35
7.2 - Framework CakePHP.....	36
7.3 - Servidor Amazon Web Services .....	38

7.3.1 - LAMP .....	40
7.4 - Base de dades MySQL.....	41
7.5 - Git .....	41
7.6 - Repositori Bitbucket.....	41
7.7 - Trello board .....	43
7.8 - IDE Eclipse.....	43
7.9 - Tècniques de recomanació.....	44
7.10 - Llibreria Apache Maths .....	45
7.11 - ElasticSearch .....	46
7.12 - <i>Dataset</i> : Esdeveniments i Usuaris .....	47
8. Anàlisi i disseny del sistema.....	48
8.1. Diagrames de casos d'ús .....	48
8.2 - Fitxes de casos d'ús.....	50
8.3 Diagrames de seqüència .....	58
8.4 - Diagrames de classe .....	60
8.4.1 - Diagrama del Sistema Recomanador.....	60
8.4.2 - Diagrama de l'aplicació web Boonobo .....	62
8.5 - Diagrama d'Entitat-Relació (Base de dades) .....	63
9. Implementació i proves .....	66
9.1- Estructura del sistema recomanador .....	66
9.2 - Els apartats de l'aplicatiu web .....	69
9.3 - Fases de proves del funcionament.....	69
10. Implantació i resultats .....	70
10.1 - Implantació.....	70
10.2 - Resultats .....	70
10.2.1- Login.....	70
10.2.2- Registrar-se .....	72
10.2.3- Llegir termes i condicions.....	75
10.2.4- Llegir/acceptar política de cookies .....	75
10.2.5- Tancar sessió .....	76
10.2.6- Visualitzar esdeveniments .....	76
10.2.7- Visualitzar esdeveniments recomanats .....	78
10.2.8- Visualitzar Top 20 d'esdeveniments.....	78
10.2.9- Afegir amistat.....	79
10.2.10- Editar perfil.....	80

10.2.11- Visualitzar esdeveniment .....	83
10.2.12- Xatejar amb usuaris .....	85
10.2.13- Afegir comentaris en un esdeveniment .....	86
10.2.14- Cercar usuaris.....	88
11. Conclusions o assoliment dels objectius .....	89
11.1 - Conclusions derivades dels resultats.....	89
11.2 - Conclusions derivades de la implementació .....	95
11.3 - Compliment de la legislació vigent.....	98
12. Treball futur tecnologies o aplicacions existents.....	99
13. Bibliografia .....	101
14. Annexos.....	103
14.1 - Annex 1: document enviat als futurs usuaris per testar l'aplicació .....	103
14.2 - Annex 2: formulari google per recol·lectar feedback del sistema recomanador ...	105
14.3 - Annex 3: plannificació de les tasques dels sprints .....	107

# 1. Introducció, motivacions, propòsit i objectius del projecte

IBM va afirmar després d'un exhaustiu estudi<sup>1</sup> que el 90% de les dades d'Internet s'havien produït durant l'últim any (referint-se al 2014) i que el 80% d'aquestes venien de la gran aglomeració de xarxes socials que havien sorgit.

Els sistemes recomanadors (veure apartat 5.3) han proliferat en aquestes xarxes com a ens generador de noves relacions socials o com a nucli central d'aplicacions en tant que són capaços de proporcionar a l'usuari un conjunt de nous ítems a ser descoberts afins al seu comportament. La importància doncs recau en com capturar, processar i analitzar aquest comportament.

Aproximadament un any abans d'iniciar aquest projecte, un grup de cinc amics decideixen muntar una startup<sup>2</sup> tecnològica que tingui com a principal motivació desenvolupar tota mena de projectes i idees que els socis tinguin. La més consolidada, i la que fins al dia de la redacció d'aquestes línies és la que més madura està, és la d'una xarxa social d'esdeveniments. La idea surt en la necessitat d'explotar un àmbit tant poc tractat per la gran xarxa social com és Facebook. Aquesta permet fer una gestió d'esdeveniments completa per part de l'usuari però, mentre contínuament la plataforma utilitza sistemes col·laboratius per recomanar possibles amistats, aplicacions i pàgines promocionades, els esdeveniments són la gran baula perduda. Les xarxes socials regeixen el comportament dels milions d'usuaris que estan a Internet, són les pautes per la nova sociabilitat i les tendències actuals. L'ésser humà sempre ha tendit a relacionar-se en grups de gent com podria ser la família, els companys de feina, els amics de la universitat... s'han construït xarxes interpersonals que han influenciat l'evolució global d'aquests col·lectius. Les xarxes socials virtuals han deixat de ser una plataforma purament tecnològica per ser l'ens d'estudi de la psicologia, antropologia, economia, ciències socials... Els sistemes recomanadors són un dels softwares que interpreta les dades que es recullen d'aquestes amb la finalitat d'ampliar les connexions que existeixen en aquesta xarxa de manera eficient.

Actualment existeixen grans xarxes socials de tota mena d'activitats, es classifiquen en tres grans grups segons el seu objectiu principal:

- Xarxes socials genèriques (les més nombroses i sense un objectiu principal a part de d'interconnectar persones: Facebook, Twitter, Google+...)

---

<sup>1</sup> Estudi de Big Data generat per IBM - <https://www-01.ibm.com/software/data/bigdata/what-is-big-data.html>

<sup>2</sup> Startup: terme utilitzat per referir-se a empreses incipients o emergents, amb poc temps de vida.

- Xarxes socials professionals (creades per aproximar especialistes en sectors i relacionar la demanda amb l'oferta: LinkedIn, JobAndTalent..)
- Xarxes socials verticals o temàtiques (basades en un tema en concret com pot set activitat, hobbie o rol: Instagram, Snapchat, Youtube, Pinterest..)

Nosaltres volem convertir-nos en el referent de xarxa social on un usuari hi busqui activitats per cobrir el seu temps lliure. On no calgui anar a diferents plataformes i aplicacions de temàtiques diferents per trobar quina serà la planificació del cap de setmana, per exemple.

A partir d'un estudi de viabilitat per part de la incubadora que acull el projecte, s'inicien les pautes per desenvolupar aquesta aplicació i se la bateja amb el nom de Boonobo. El nom (com a curiositat) prové d'una raça de ximpanzé que se la coneix per ser la més semblant a l'ésser humà ja que posseeix el sentiment d'identitat pròpia i a més es relaciona i socialitza per gust, no per necessitat bàsica. Aquesta és la idea essencial que volíem que tingués la nostra aplicació.

En aquest projecte, el concepte de "sistema recomanador social" es defineix com la combinació de la informació d'una xarxa social, que pot variar segons el nostre comportament personal a la web, i la interacció que realitzem amb tots els usuaris de la plataforma. Amb el desenvolupament de les xarxes socials, com utilitzar la informació que n'extraïem ha estat un tòpic molt estudiat per desenvolupar aplicacions en *background*. La vida de l'ésser humà està constantment influenciada pel seu entorn i per la societat on en troba, trobar aquests patrons d'influència i explicar-ne els seus motius és el principal objectiu d'aquesta ciència. En aquest projecte s'exposa una de les possibles vies per resoldre aquest problema, però no n'és l'única: hi ha molts algorismes que intenten aproximar les recomanacions d'un sistema artificial a les que es donarien en un entorn de persona a persona. Cal però identificar inicialment quins seran els productes o *items* a recomanar, quines seràn les condicions de la xarxa social i quina informació obtindrem de l'usuari. Seran aquests tres, condicionants de les decisions que es prenguin en el desenvolupament del Sistema Recomanador.

## 1.1 - Motivacions

La principal motivació que m'ha portat de desenvolupar aquest projecte és l'aprenentatge en el món dels sistemes recomanadors que suposa. És un món dins de la intel·ligència artificial que sempre m'havia despertat interès i aquesta és la oportunitat perfecte per fer-ne estudi, investigació i a més deixar un programari llest pel seu ús. També ha estat important per a mi

desenvolupar un projecte sorgit d'una idea pròpia o, si més no, d'un col·lectiu on hi formo part. He hagut de marcar jo les pautes, el *workflow* i l'abast del projecte i això ha estat tot un repte acadèmic i professional. I per últim les ganes de conèixer amb més profunditat el funcionament de tot un conjunt de xarxes socials que utilitzem diàriament desconeixent un dels seus *back-sistemes* com és el sistema recomanador.

## 1.2- Propòsits i objectius

El projecte consisteix en construir un sistema recomanador que basant-se en l'activitat dels usuaris en l'aplicació, recomani un llistat d'esdeveniments a aquest. El projecte s'engloba dins de l'àmbit de la intel·ligència artificial, més concretament els recomanadors col·laboratius que explicarem més endavant. Ens basarem en diferents maneres d'enfocar les dades per obtenir esdeveniments amb procediments diferents. Per aconseguir-ho dividirem aquest projecte en tres pilars:

- El desenvolupament d'una aplicació web per poder interaccionar amb l'usuari i no només recollir les dades necessàries d'aquest, sinó mostrar els resultats i recollir el feedback. Aquesta permetrà registrar-se, loguejar-se, establir amistats amb altres usuaris, buscar usuaris, buscar esdeveniments, filtrar esdeveniments, mostrar esdeveniments i establir el vincle entre l'usuari i l'esdeveniment.
- La implementació d'un algorisme que a partir de les dades recollides en l'aplicació, ens calculi aquestes recomanacions. Aquest algorisme tindrà diferents bases que es descriuen més endavant i a les quals es vol assolir en totalitat. Basant-nos en el sistema col·laboratiu, s'implementarà paral·lelament un model *Nearest Neighborhood* i un model *All Neighborhood* i s'aniran complementant amb nous filtres sorgits de la metodologia escollida per desenvolupar el projecte.
- La recollida del *feedback* dels usuaris després de l'ús de l'aplicació amb el sistema recomanador i l'elaboració d'unes estadístiques de funcionament. Es demanarà a un set real d'usuaris que valorin la seva experiència en l'aplicació en diferents escenaris diferents. La única manera d'elaborar unes conclusions sobre un sistema recomanador és a través del *feedback* que el propi usuari dona d'aquest.

Tot i que el projecte com a tal no es desenvolupa en cap entorn consolidat, sí que parteix d'un projecte iniciat com s'ha comentat anteriorment. El projecte Boonobo cedeix la captura d'esdeveniments reals realitzats a diferents webs d'esdeveniments i captures de Facebook

com a *data-set* inicial. També cedeix el diagrama Entitat-Relació de la Base de dades des d'on parteix el disseny de l'aplicació web que serà complementat amb els nous camps i taules necessaris pel sistema recomanador. Per últim cedeix la imatge gràfica com és el logotip, tipografia original, paleta de colors i plantilla de bootstrap (modificada també durant la realització d'aquest projecte) que s'utilitzarà en el definitiu.

## 2. Estudi de viabilitat

Aquest projecte serà desenvolupat únicament per l'estudiant i les eines de les que ja disposava, pel que podem deduir que el cost final d'aquest projecte ha estat reduït. Cal avançar que l'estructura física inicial del projecte no serà la mateixa que la final, però tot i així el cost serà el mateix.

Inicialment utilitzarem un ordinador qualsevol, amb qualsevol sistema operatiu que tingui un AMP<sup>3</sup> instal·lat (Apache+MySQL+PHP) , un JDK-JSE8 amb servidor Tomcat i un sistema ElasticSearch/Lucene per indexar; per treballar amb el codi necessitarem crear un repositori git local.

El projecte estarà allotjat en dos repositoris Bitbucket privats i clonats en els diferents dispositius que s'utilitzarà per programar.

Seguidament, el projecte serà allotjat en un servidor AWS (Amazon Web Service) totalment gratuït perquè sigui accessible, de manera segura, des de qualsevol punt de la xarxa. Facilitant així el registre d'usuaris i la demostració final. Aquest té un cost de 0€ el primer any d'allotjament en servei bàsic.

Anteriorment hem esmentat el programari bàsic pel correcte funcionament de l'aplicació, totes elles són eines gratuïtes a les que també afegim les utilitzades durant el desenvolupament:

- IDE Eclipse
- SourceTree
- Git Bash
- Mysql Workbench
- Framework CakePHP
- Trello board

En tot cas podríem afegir el cost del programador, però al ser un PFG no inclòs en empresa aquest cost seria també de 0€

---

<sup>3</sup> Depenent del sistema operatiu estarem parlant de WAMP (Windows), LAMP (Linux), MAMP (Mac OS)

### 3. Metodologia

En el projecte s'ha seguit una metodologia de desenvolupament basada en el *Agile software development* (figura 1) ja que era la que més s'adaptava a les necessitats del projecte. Les característiques diferencials en aquest cas és que en el cas de l'aplicació web s'ha caracteritzat per iteracions (anomenades *sprints*) setmanals i en el cas del sistema recomanador en iteracions variables que duraven fins a tres setmanes.

En cada una de les iteracions s'ha establert un *workflow* de treball i s'han designat unes tasques que han estat constantment documentades en un *board* de treball; aquest s'explica en el següent apartat de planificació.

En ser un projecte d'investigació amb un objectiu clar, però no un camí fix, a mesura que s'anava desenvolupant sortien moltes idees (en el cas del sistema recomanador) i funcionalitats (en el cas de la interfície de la *web-app*) i la metodologia àgil ens ha permès incloure-les sense problemes.

Les iteracions han estat:

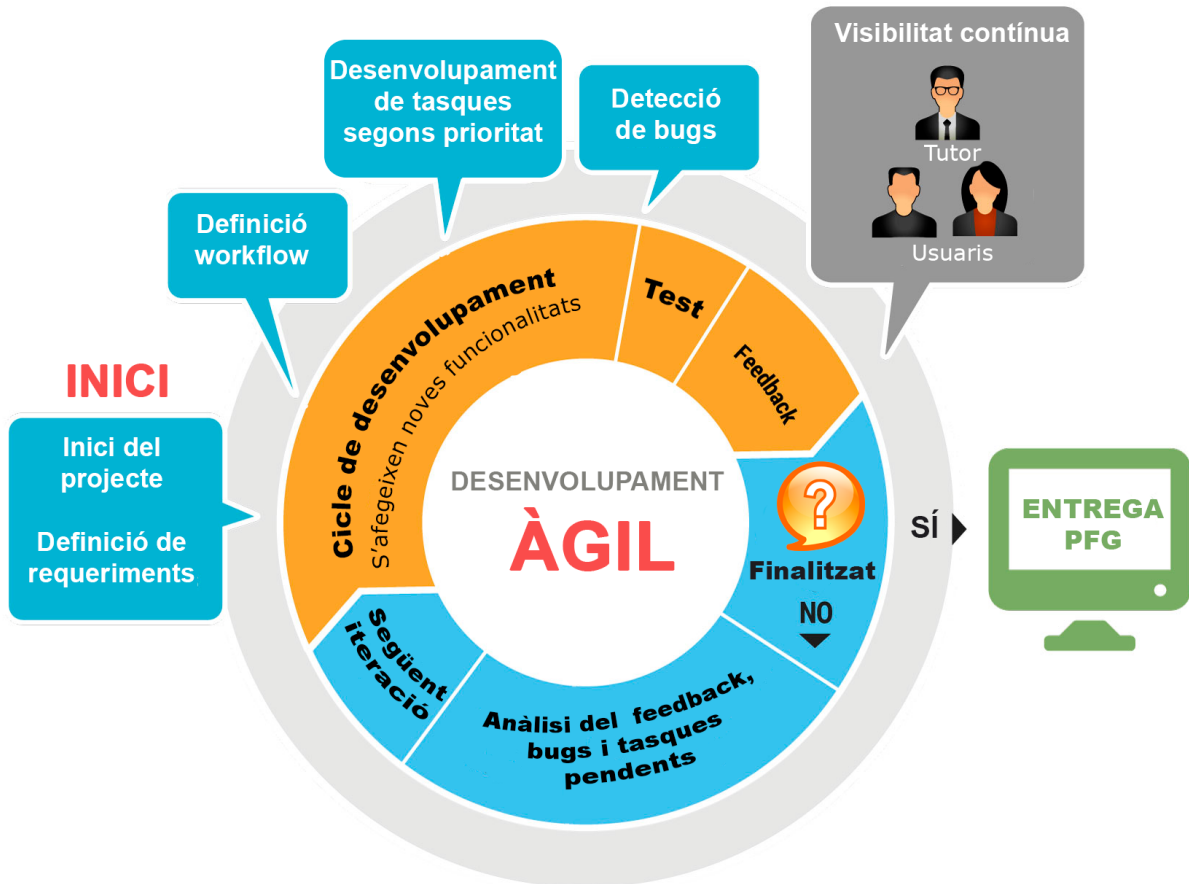
1. Conèixer l'entorn
2. Decidir el llenguatge de programació i eines a utilitzar
3. Estudiar les eines escollides
4. Estructuració del treball segons les prioritats i les funcionalitats
5. Desenvolupament del programari a base de *sprints*. Aquest ha estat un procés iteratiu que una vegada finalitzada cada iteració, si el projecte no estava finalitzat o no s'havia arribat a la *due date*<sup>4</sup>, es reiniciava.
  - 5.1. Generació d'un *backlog* de tasques segons la funcionalitat a cobrir per la iteració.
  - 5.2. Tria de les tasques a realitzar en la iteració segons la prioritat dins el nostre projecte
  - 5.3. Disseny de les tasques i divisió de desenvolupament per cobrir totes les tasques triades
  - 5.4. Codificació de les tasques assignades
  - 5.5. *Testing* per verificar el bon funcionament de les tasques, si es detecta algun error, es classifica com a "*Bug*", se li assigna prioritat màxima i formarà part del punt 5.2 de la següent iteració.
  - 5.6. Documentació de la iteració realitzada

---

<sup>4</sup> En aquest cas estariem parlant de la data d'entrega del projecte, no d'una data preestablerta per algun client.

5.7. Feedback dels usuaris, tutor o propi estudiant. Si el projecte no està finalitzat, retorn al punt 5.1

6. Finalitzar la documentació



Il·lustració 1 - Cicle agile de l'aplicació

## 4. Planificació

La idea d'aquest projecte sorgeix en el moment en que una aplicació que estava en desenvolupament tenia un seguit de tasques que requerien més temps que altres. Una d'elles, el sistema recomanador d'esdeveniments, em va fer despertar el interès en saber-ne més d'aquest món així que vaig decidir desenvolupar -lo com a projecte de final de grau amb la condició que no seria propietari i confidencial, sinó que es registraria sota llicència de codi GPL 3.0 <sup>5</sup>. Des del inici, gener era la data escollida d'entrega.

<sup>5</sup>GNU General Public License: se'n permet ús, modificació i comercialització però qualsevol modificació ha de distribuir-se en llicència GPL

Les primeres setmanes es van dedicar a acotar el projecte i a seleccionar les dades del projecte que s'utilitzarien en el desenvolupament d'aquest PFG. També es van dedicar a seleccionar la tecnologia que s'utilitzaria en aquest i a planificar el temps d'aprenentatge dels diferents softwares i conceptes.

- Una de les eines principals utilitzada és el *framework cakePHP*, tot i que s'hi havia treballat anteriorment, s'ha requerit temps addicional per aprendre a configurar-lo en totalitat.
- El *hosting* de l'aplicació web i del sistema recomanador ve proporcionat per *Amazon* i els seus *cloud services*. S'ha necessitat un temps d'aprenentatge de l'ús i configuració d'aquests serveis.
- Una de les llibreries que utilitza el sistema recomanador per realitzar els càlculs sobre els vectors *user-event* és *rJava*, llibreria de modelació estadística que facilita els càlculs en la *JVM*.



II-lustració 2 - Agile board

L'octubre va ser dividit en quatre *sprints* setmanals que han englobat el procés de desenvolupament de l'aplicació web. Aquests *sprints* es van planificar sobre unes funcionalitats bàsiques d'una xarxa social i la incorporació de noves funcionalitats finals i *bugs* al *backlog*. Es va iniciar la fase de disseny i requeriments de l'aplicació i es van realitzar els esbossos dels diagrames de casos d'ús i fitxes partint del diagrama de classes simplificat el que heretàvem de

l'aplicatiu mestre<sup>6</sup>.

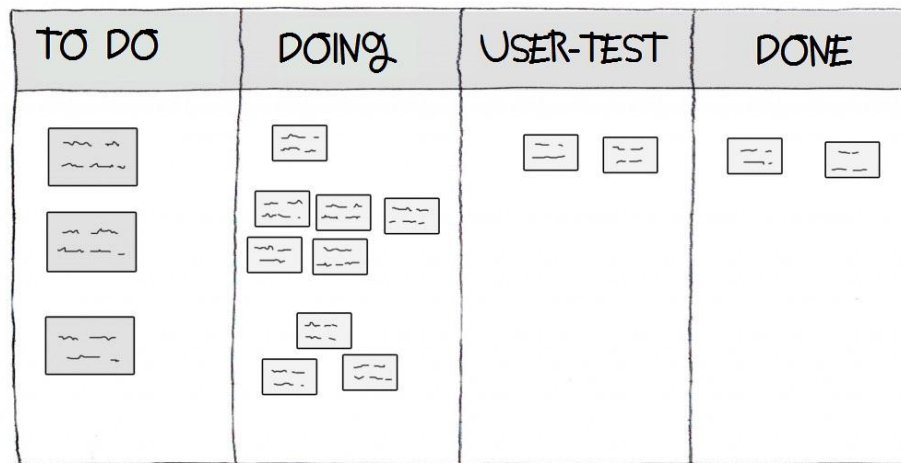
El novembre ha constatat d'un procés iteratiu format per dos tipus de *sprints*: de l'aplicació web i del sistema recomanador. L'objectiu era tenir l'aplicació en *cloud* funcionant amb una primera versió del sistema recomanador.

El desembre s'ha caracteritzat per ser un model típic de desenvolupament àgil on s'han anat incorporant de manera modular noves característiques i fix *bugs* sobre un software ja funcionant. Parlem de característiques i no funcionalitats, ja que no afegeixen nous

<sup>6</sup>En qualsevol moment que parlem d'aplicació mestre o principal, es referirem al software pel qual realitzem aquest PFG, l'inicial des d'on partim

comportaments al sistema recomanador, sinó que en milloren la precisió de la sortida. Seria una altra manera d'interpretar els *issues*<sup>7</sup> que es descriuen en un *backlog*.

S'ha optat per una eina com Trello per realitzar un *board* semblant al que ens deixaria *Scrum*: iteracions constants basades en funcionalitats variables, però no inesperades. Les columnes triades per treballar són variables segons la magnitud del treball, en aquest cas tenim :

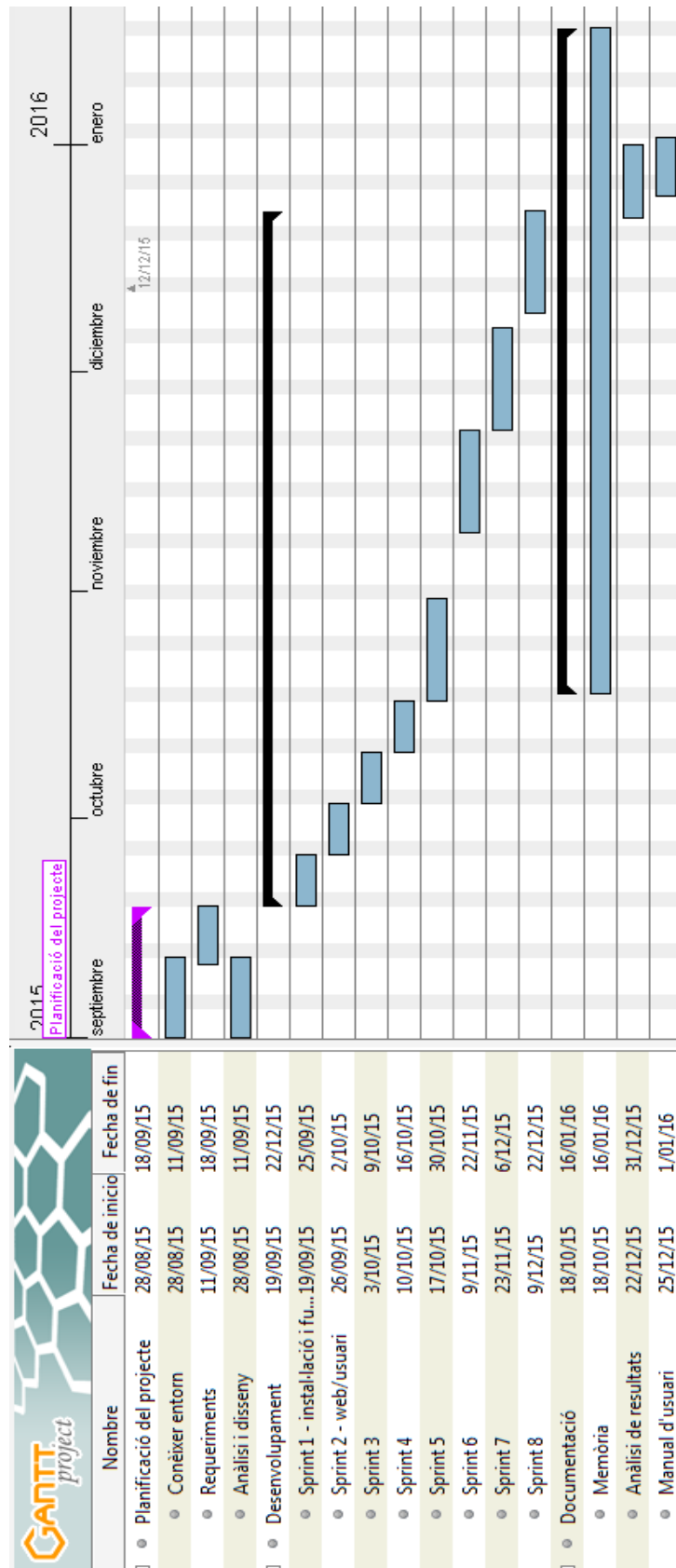


Il·lustració 3 - columnes del nostre board

La documentació s'ha inclòs en cada *sprint* realitzat (un apartat per *sprint*) i la resta d'apartats s'han redactat un cop finalitzat el desenvolupament del programari. Per il·lustrar millor el procés de realització del PFG, s'ha realitzat un diagrama de Gantt, amb l'eina *GanttProject*<sup>8</sup>, que és un projecte de software lliure i gratuït (figura 3).

<sup>7</sup> Cadascuna de les tasques esmentades en un *backlog*

<sup>8</sup>"GanttProject: free desktop project management app." 2006. 23 Jul. 2015 <http://ganttproject.biz/>



II-lustració 4 - Diagrama de Gantt de la planificació del projecte

## 5. Marc de treball i conceptes prèvis

En aquest apartat s'expliquen els conceptes necessaris per a entendre els algorismes desenvolupats i les decisions preses per desenvolupar aquest projecte. També s'expliquen aspectes a tenir en compte per entendre el procés de gestació d'aquest projecte i la idea inicial.

### 5.1 - Xarxa social

Un servei de xarxa social és un mitjà de comunicació social que es centra en establir un contacte amb altres persones per mitjà d'Internet. Estan formades per un conjunt d'equips, servidors, programes, conductors, transmissors, receptors i sobre tot per persones que comparteixen alguna relació, principalment d'amistat, on mantenen interessos i activitats en comú o es troben interessats en explorar els interessos i les activitats d'altres usuaris.

A mitjan de l'any 2009, les empreses (sobretot europees) les van començar a explotar com un mitjà per desfogar-se publicitàriament i crear perfils de consumidors segons la seva activitat.

En general, els serveis de xarxes socials permeten als usuaris crear una compte d'usuari amb un perfil que els identifica, el qual haurà de incloure una certa informació personal de l'usuari. Un cop accedeixen al seu compte, podran establir contacte amb altres individus i després els serà possible comunicar-se entre ells. Encara que la majoria de les xarxes incorpora la opció de carregar imatges personals, certes xarxes estableixen un protocol d'imatges com a avatar que obliga als usuaris a identificar-se amb quelcom més. De xarxes socials en tenim dos tipus, les privades (com podrien ser les internes en una empresa) i les ESN o xarxes socials obertes on està inclòs aquest projecte. Els usuaris amb freqüència poden ser amics d'altres usuaris, encara que no es coneixin prèviament.

En la majoria dels serveis de xarxes socials, davant una sol·licitud o invitació per establir contacte virtual, els usuaris han de confirmar que són amics i que accepten aquesta connexió. Una altra característica de les xarxes socials és l'adaptació del terme principal que relaciona als usuaris amb valoracions i puntuacions a aquests ítems. En trobem exemples a *Facebook*, *Twitter*, *Instagram*... inclòs plataformes web com *Amazon*, que ha volgut incloure el terme social al seu principal objectiu que seria vendre (per suposat per afavorir-lo).

De xarxes socials en tenim molts tipus i ambientades a diferents àrees i actualment estan enfocades a relacionar activitats quotidianes i trencar la barrera de la privacitat diària dels usuaris.

## 5.2 - Data analytics & predictive analytics

*Data analytics* o anàlisi de les dades és el procés d'inspeccionar, netejar, transformar i modelar dades amb l'objectiu de descobrir informació útil, suggerir conclusions i recolzar la presa de decisions. L'anàlisi de dades té varies facetes i enfocaments compassant varies tècniques amb gran varietat de noms en finances, ciència i ciències socials.

El *data mining*, per exemple el utilitzat en l'obtenció d'esdeveniments previs a aquest projecte, és un tipus d'anàlisi de dades que s'enfoca en modelar i descobrir coneixement predictiu amb finalitat descriptiva.

No obstant, la tècnica d'anàlisi de dades utilitzada sobretot en aquest aplicatiu és l'anàlisi predictiu que està enfocat a construir models estadístics i classificació predictiva .

L'anàlisi predictiu es val de valors estadístics per analitzar dades actuals i històrics, per fer prediccions sobre esdeveniments futurs. Aquestes prediccions no són afirmacions absolutes, sinó que són mètriques de probabilitat de que certs esdeveniments passin de debò en un futur. L'efecte funcional que busquen aquestes iniciatives tècniques és que l'anàlisi predictiu doni una puntuació (una probabilitat com hem dit) per a cada subjecte de l'anàlisi (en aquest cas un usuari) i un ítem específic (els esdeveniments) amb el fi de determinar, informar o influir processos en la plataforma de forma massiva. D'anàlisi predictiu en tenim de tres tipus: model predictiu, model descriptiu i model de decisió. No obstant, per la naturalesa del projecte ens basarem en el predictiu.

Els models predictius són models de la relació entre el rendiment específic d'un subjecte en una mostra i un o més atributs o característiques d'aquest subjecte. L'objectiu del model és avaluar la probabilitat de que un subjecte semblant tingui el mateix rendiment en una mostra diferent i poder trobar patrons de comportament que responguin per a futures mostres i que preestableixin relacions. Podem veure però, que d'aquests models n'hi ha de tant avançats com a mostres de *marketing* i borsa com la simple relació d'aquest subjecte amb productes o esdeveniments, tal com es dona en un sistema recomanador. No és d'estranyar que aquests models ofereixin molts avantatges:

- Processos que requereixen d'un gran número de decisions semblants o aplicables segons el mateix patró.

- Accions que tinguin impacte significatiu, en termes econòmics o de sostenibilitat del negoci.
- Estudi del comportament dels usuaris segons perfils interns del sistema.
- Tractament de les dades que generen els usuaris al navegar per la plataforma del sistema recomanador.
- Analitzar l'efectivitat d'uns ítems vers altres (esdeveniments) segons antics i futurs comportaments i dades
- El mercat actual és tan dinàmic que cal fer un pas endavant al perfil de consumidor que modelen.

Però actualment els sistemes recomanadors lluiten contra un gran desavantatge que es presenta de maneres diferents segons el territori: l'ètica. Aquesta és diferent segons la legislació estatal o fins i tot comunitària com seria el cas de la Unió Europea o els Estats Units. En cada un d'aquests estats es permet la recol·lecció i tractament de diferents dades dels usuaris i això implica un canvi de paradigma en els sistemes recomanadors.

### 5.3 - Sistema Recomanador

Tradicionalment la gent ha utilitzat gran varietat d'estratègies per a resoldre els problemes de la presa de decisions: converses amb amics, obtenir informació d'un col·lectiu sobre el que ens fiem, la contractació d'experts, consulta a Internet.. o simplement seguint a la multitud. Tanmateix, tots hem experimentat moments en que els consells o gustos d'un amic no s'han correspost amb els nostres. O situacions en que després de moltes hores d'Internet i diferents opinions, ens han creat més aviat una situació de confusió.

Realment un bon consell és difícil de rebre i difícil de valorar ja que només sota el nostre punt de vista, quelcom que sigui recomanat serà adequat 100% per a nosaltres. No seria increïble doncs aproximar-nos a aquest 100% i trobar algú que ens fens recomanacions eficients?

La construcció de sistemes que acompanyen a l'usuari en la seva presa de decisions és una de les metes dels sistemes recomanadors. Sobretot, en una recomanació de qualitat buscant en primer lloc l'eficiència.

El repte de proveir una recomanació assequible, personal i de màxima qualitat és l'eix central dels dos pilars d'aquest sistema que els dividirem entre els aspectes tècnics i els aspectes psicològics. És a dir, parlem de tècnics quan l'objectiu és trobar regles i mètodes que explorin i explotin la informació existent d'una manera efectiva i eficient; els factors

psicològics són tots aquells que acompanyen les dades a l'hora de presentar-se davant de l'usuari i els factors que engloben a l'usuari en el moment d'entrar dins el sistema (estat anímic, caràcter, circumstàncies personals...). Ja fa més de 15 anys des de que aquests sistemes que intenten cobrir els dos pilars són anomenats "sistemes recomanadors" i si be van començar com a filtres de preferències i de casos antics, ara es proveeixen d'un seguit d'algorismes artificials que aconseguen que cadascú sigui més complex que l'anterior.

L'objectiu que persegueix aquest projecte és elaborar les anomenades recomanacions personalitzades, que per suposat és el que hauria de buscar qualsevol recomanador, terme introduït per *Netflix*<sup>9</sup>. Proveir un portal de recomanacions personalitzades però, implica que aquests sistema ha de saber-ho tot sobre l'usuari. Introduïm doncs una primera idea que seria el perfil d'usuari o el model usuari que genera un sistema recomanador per a poder treballar sobre aquest usuari. No cal que sigui complex, per començar podria tenir per exemple les preferències i gustos de la persona. La construcció d'aquest model és clau per al sistema recomanador i la manera de construir-lo depèn de cada aplicatiu. Quant a les preferències de l'usuari hi ha dues maneres d'obtenir-les: de manera explícita o implícita. La primera, és demanar clarament a l'usuari que d'alguna manera entri les seves preferències segons els ítems que recomanarem (característiques, necessitats, categories...); la segona és la captura del comportament de l'usuari en l'aplicatiu per construir-ne un model: quines pàgines visita, quant de temps les visita, quines són les que visita més, quina visita després d'una altra... i podem obtenir d'una manera no directe els gustos de l'usuari.

Per a construir un bon model d'usuari ens els aplicatius actuals però, ens falta un últim pilar: la comunitat. Com hem introduït anteriorment, les xarxes socials i les noves tecnologies aporten als consumidors un seguit de comunitats de les que tots acabem formant part des del nostre *smartphone* o ordinador. Per tant cal tenir en compte aquest entorn de presa de decisions (entorn on el nostre sistema recomanador treballarà per obtenir dades i dibuixar patrons). Aquest entorn comunitari ens proporcionarà informació addicional com les amistats del nostre usuari i per tant, les preferències d'aquestes amistats. Construirem doncs el nomenat *neighborhood* o mostra de veïns que podrà anar des de tot el conjunt d'usuaris fins a subgrups que realitzi el propi usuari de les seves amistats.

Depenent de quines preferències utilitzem per treballar, com les hem aconseguit i com construir l'entorn de recollida de dades, distingim diferents sistemes recomanadors. Cada classificació va començar sent una millora de l'anterior fins al dia d'avui, on tot sistema recomanador actual implementa un híbrid de tots ells per triar com a model el millor perfil d'usuari. Segons les preferències que coneixem de l'usuari parlaríem de *Content-based filtering* (filtre basat en contingut), segons les decisions passades de l'usuari i el seu

<sup>9</sup> Portal i xarxa social per a veure i recomanar pel·lícules, es va convertir en pioner dels sistemes recomanadors des de la seva creació: <https://www.netflix.com/es/>

comportament davant dels ítems de la plataforma tindriem el *Knowledge based filtering* (filtre basat en l'experiència) i per últim el que estableix els paradigmes de la col·laboració i comunitat seria el *Collaborative sistem* (sistema col·laboratiu). Els dos primers, és curiós com amb els anys s'han identificat més com a filtres que com a sistemes intel·ligents, propietat que sí rep l'últim tipus de sistema recomanador. Però com hem comentat, actualment grans plataformes com *Facebook*, *Netflix*, *Amazon*... implementen sistemes híbrids jugant amb els pesos de cadascú d'aquests.

Per tal de proporcionar com més possible i millor la característica "d'intel·ligent" a aquest projecte, s'ha optat per fer un híbrid en un únic sistema semblant als que implementen les grans plataformes socials.

### 5.3.1 - Sistema Col·laboratiu

La idea principal de l'elaboració d'una recomanació col·laborativa és explotar la informació sobre el comportament anterior o les opinions dels usuaris existents en la comunitat per a predir quins ítems o productes probablement més li agradarà o no a l'usuari actual.

Aquests sistemes són els últims que van aparèixer en el món dels sistemes recomanadors i, tot i que tenen limitacions palpables, descriuen més a la perfecció el procés d'elecció i recomanació segons el nostre raonament.



**Il·lustració 5 - Esquema del sistema col·laboratiu**

Per enfocar-ho i fer-nos una primera idea de com funciona, imaginem-nos una matriu d'usuaris i ítems i com a valors, per exemple, les seves avaluacions o ràtings sobre els productes que volem estudiar. És important tenir clar aquest ràting: la sortida del nostre sistema recomanador tindrà la mateixa base que el estàndard utilitzat per a construir els nostres ràtings. Si per exemple, parlem de puntuació de llibres que van de 1 a 5, la predicció

de la nostra recomanació estarà dins d'aquest model. Els models es poden basar en ràtings binaris o ràtings extrems (negatius pels casos no desitjats i positius pels casos desitjats). A partir d'aquesta entrada obtindrem la sortida que tant pot ser un sol producte recomanat, com una llista de productes que, per suposat, cap d'ells haurà estat consumit prèviament.

	Item1	Item2	Item3	Item4	Item5
Alice	5	3	4	4	?
User1	3	1	2	3	3
User2	4	3	4	3	5
User3	3	3	1	5	4
User4	1	5	5	2	1

**II-l·lustració 6 - Exemple de matriu de ratings**

Veiem en la figura 6 com tenim un usuari sobre el que volem treballar, Alice, i ens preguntem quin seria el ràting que podria tenir sobre el ítem 5 (imaginem que pot ser un llibre, una pel·lícula...) sobre el valor que obtinguem sabrem si serà aconsellable o no recomanar-li aquest producte.

Hi ha moltes maneres d'avaluar aquesta matriu, es pot fer treballant des del punt de vista dels productes o des del punt de vista dels usuaris. Òbviament, nosaltres continuarem l'exemple fent servir el mateix procediment que s'ha utilitzat per desenvolupar el SR d'aquest projecte. Primer de tot necessitem calcular la similitud d'Alice amb la resta d'usuaris, que ara anomenarem veïns i més endavant explicarem els tipus abans esmentats. Aquesta similitud es pot calcular de moltes maneres, nosaltres hem apostat per la fórmula de la Correlació de *Pearson*, que no és la millor manera però sí que és aquella que proporciona un resultat dins el estàndard d'estudi i dins el domini esperat.

$$stm(a,b) = \frac{\sum_{p \in P} (r_{a,p} - \bar{r}_a)(r_{b,p} - \bar{r}_b)}{\sqrt{\sum_{p \in P} (r_{a,p} - \bar{r}_a)^2} \sqrt{\sum_{p \in P} (r_{b,p} - \bar{r}_b)^2}}$$

a i b són els usuaris,  $r_{a,p}$  és el ràting que relaciona a l'usuari 'a' amb el producte 'p' i aquest està dins la col·lecció P que representa tots els ítems.

El resultat de la correlació ens donarà una similitud que anirà des de -1 (fortament negativa) a 1 (fortament positiva).

Seguidament caldrà calcular la predicció directa de l'usuari Alice per l'Item5, per aconseguir-

ho utilitzarem la següent fórmula (bastant comú en prediccions estadístiques):

$$pred(a, p) = \bar{r}_a + \frac{\sum_{b \in N} sim(a, b) * (r_{b,p} - \bar{r}_b)}{\sum_{b \in N} sim(a, b)}$$

On ens nodrim de totes les semblances calculades d'Alice amb la resta d'usuaris, la predicció de la resta d'usuaris per a l'ítem p que estem mesurant ( $r_{b,p}$ ) i la mitjana de ràtings de l'usuari b amb qui estem comparant ( $\text{mean}(r_b)$ ).

La utilització de la Correlació de *Pearson* en aquest cas podria ser un dels motius de discussió del projecte, és cert que no és la millor fórmula per calcular els usuaris amb qui més semblança tindrà Alice (hi ha d'altres com la semblança per cosinus ajustat o el mètode *Spearman's*). El problema és que si en el nostre domini tenim un seguit d'ítems els quals són agradats per a la majoria, amb *Pearson* els acabarem recomanant a tothom. Això és perquè *Pearson* no té en compte la polèmica de que dos usuaris els hi agradi un ítem poc comú, hauria de comptar més que no si estan d'acord en una puntuació sobre un ítem que agrada a tothom. Hi ha moltes solucions que es poden aplicar en aquest punt, però no significa que resolguin en millors recomanacions. S'aconsella doncs complementar *Pearson* amb els anomenats pesos significatius, que s'abstreuen de *Content-based* i el *Knowledge based*.

Les preguntes de quin estàndard fem servir per decidir a partir de quant ho incloem a la llista de recomanacions o què passa amb aquells ítems que encara no tenen avaluacions per part d'altres usuaris, ho anirem responent en aquest document a mesura que ens endinsem en el projecte en sí.

### 5.3.2 - Nearest Neighborhood i All Neighborhood

Quan dibuixem la matriu que volem emplenar, és important decidir quins són els usuaris que inclorem en la llista a part de l'usuari que estem avaluant. Amb aquests, després farem perfils de semblança i seran decisius per a la construcció de prediccions cap a l'usuari. Intuïtivament un pensa que és millor fer una selecció dels usuaris que utilitzarem, per al càlcul de prediccions utilitzarem aquells que tinguin només una correlació positiva amb el nostre usuari (*Nearest Neighborhood*) i per suposat que hagin puntuat l'ítem que estem avaluant. Si incloem tots els usuaris en aquesta predicció no te perquè afectar de manera negativa al resultat, però segur que ens afectarà al temps d'execució i a la precisió de la predicció.

La selecció de veïns és un altre dels grans pilars d'estudi dels sistemes recomanadors, fent-ne variar molt el resultat final. Hi ha tècniques que diuen que sobre els més semblants a l'usuari, només se n'ha d'agafar un nombre constant  $k$ , el problema és que quan més petit sigui  $k$ , menys prediccions podrem fer partint que molts ítems només seran avaluats per un sol usuari o, ans el contrari, hi haurà molts ítems que se'ns recomanaran de manera incorrecta. Tindríem doncs un altre problema: com obtenim aquesta  $k$  i sobre quins usuaris l'apliquem?

Una altra tècnica introdueix els amics com a veïns més pròxims, aquells que estableixen una relació fora de l'aplicatiu, establiran una relació dins de l'aplicatiu amb l'usuari de manera que el més fàcil és basar-nos en la similitud del nostre usuari amb aquests.

S'ha optat en l'aplicació a fer els dos casos d'estudi: què passa quan la recomanació es basa en els veïns obtinguts a partir de l'amistat, o obtinguts a partir de les millors  $k$  similituds; sent  $k$  un nombre entre 20 i 50 com recomanen Herlocker i Manch<sup>10</sup> en el seu paper.

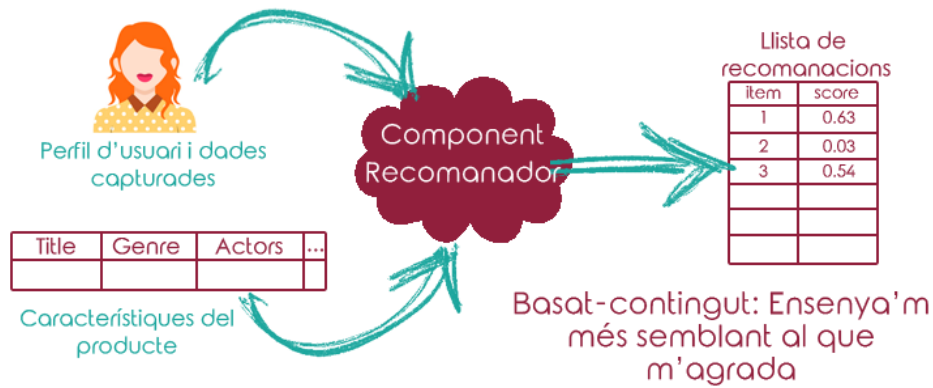
### 5.3.3 - Filtratge basat en el contingut

Per aplicar tècniques de filtratge<sup>11</sup> col·laboratiu, excepte els ratings dels usuaris, res hem de conèixer dels productes que recomanem. Poden ser llibres, pel·lícules, poden tenir unes característiques o unes altres. Voler seleccionar els més recomanables segons les seves pròpies característiques no és possible: en el món real qualsevol persona que s'hagi llegit un llibre de Harry Potter se li hauria de recomanar un altre de la saga; o identificaríem que a l'usuari li agrada la fantasia. Lluny estan aquestes conclusions de les que podem abstraure del filtratge col·laboratiu. És possible fer una recomanació basant-nos en aquest criteri si disposem de la descripció i les característiques de l'ítem que estem avaluant i el perfil del l'usuari que d'alguna manera representi l'històric de gustos.

---

<sup>10</sup>"An Algorithmic Framework for Performing Collaborative ..." 2013. 29 Dec. 2015

<sup>11</sup> Diferenciarem entre el sistema en sí que serà la base en que es construirà el sistema recomanador i els filtres que s'apliquin a la sortida d'aquest sistema per obtenir una millor precisió.



**Il·lustració 7 - Esquema del sistema basat en contingut**

Per sort, les característiques i la descripció de qualsevol ítem és fàcil de trobar actualment a Internet: tot hi és present; només necessitaríem una mica de *data-mining*, per exemple el *crawling*<sup>12</sup> que s'ha realitzat abans d'aquest projecte dels esdeveniments. No és tan fàcil trobar característiques de l'usuari que sempre ens serveixin: perquè un cop t'hagi agradat un llibre no vol dir que tots els llibres que pertanyen al mateix gènere t'agradin, o perquè el llibre fos de tapa tova no vol dir que se t'hagi de recomanar tots els llibres de tapa tova.

Els processos més avançats d'intel·ligència artificial són capaços de generar similituds entre ítems de manera bastant precisa; un exemple seria el *More Like This* que implementa *ElasticSearch*<sup>13</sup> i cerca ítems semblants a partir de les indexacions que té i els camps que li especifiquem per a comparar. D'altres utilitzen processos de *machine learning* per entendre que les característiques literalment semblants no vol dir que vagin relacionades.

En el cas d'aquest projecte, s'ha utilitzat el contingut per aplicar un filtre després del sistema col·laboratiu. Un cop s'obtenen les prediccions de recomanació, es passa un filtre basat en contingut per augmentar o disminuir el valor de la probabilitat  $p(a,r)$  on  $a$  és l'usuari i  $r$  l'ítem. Tot i així, existeixen sistemes de filtratge per contingut molt diversos i complexos que amb més temps, s'haurien estudiat.

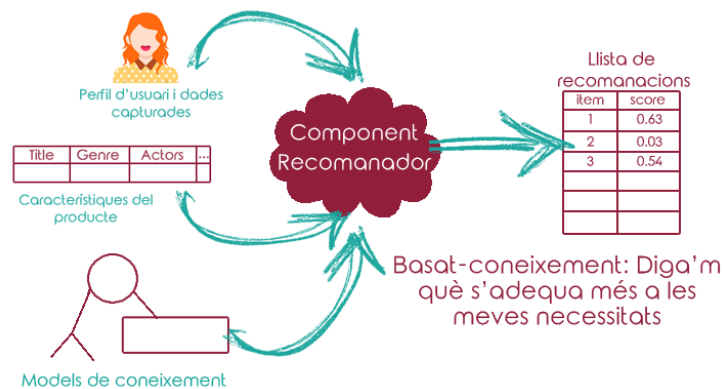
#### 5.3.4 - Filtratge basat en el coneixement

Si bé aquest filtratge no ha estat implementat en aquest projecte, es creu adequat la introducció del terme per posar-ho en context. El filtratge basat en el coneixement si bé té un objectiu similar a l'anterior explicat, el seu funcionament és diferent. L'atenció es basa en l'ítem i les seves característiques però amb la diferència que no hi entren en joc les

<sup>12</sup> Navegació i captura d'informació a partir d'una url llavor navegant pels *out-links* com una aranya fins arribar a tot Internet.

<sup>13</sup> És un servidor de cerca basat en *Lucene*. Proveeix un motor de cerca de text complet, distribuït i amb capacitat de multi-tendència.

puntuacions ni els ràtings que l'usuari pugui arribar a donar. La seva complexitat treballa per construir un model més fidel de l'usuari sabent que el nostre comportament i les nostres decisions canvien al llarg del temps segons x variables. Per exemple, si tinc 20 anys i em compro un cotxe esportiu, uns anys després em caso i tinc fills... quan vulgui anar a comprar un altre cotxe, si a banda de saber les especificacions de l'últim que vam comprar se sap la meua situació civil i familiar, la recomanació pot ser més exacta. Aquests sistemes busquen construir patrons més complexes sota el comportament humà i la nostra evolució dins una plataforma: si fa 5 anys m'agradaven les comèdies però últimament només miro pel·lícules de drama, les últimes pel·lícules haurien de tenir més pes dins el meu perfil; no perquè siguin més recents, sinó perquè les altres mostres es troben molt llunyanes en temps. Si som capaços d'imaginar l'abast d'aquests filtres de seguida ens plantejem altres fonts de coneixement de l'usuari com ara la reputació d'aquest en la xarxa, per exemple.



**II-l·lustració 8 - Esquema del sistema basat en coneixement**

El problema que planteja aquest sistema recomanador i el motiu principal pel que no ha estat implementat en aquest projecte és que cal conèixer tot un seguit de característiques dels ítems a avaluar molt específiques (i necessàries en tots, cosa que no es dona en el nostre *data-set* inicial) i necessitaríem construir un perfil d'usuari bastant complet, fet que necessita d'un ús exhaustiu de la plataforma per part dels usuaris al inici.

## 5.4 - Sistema de control de versions i Git

Un sistema de control de versions és un sistema de gestió d'arxius i directoris, que la seva principal característica és que manté la història dels canvis i modificacions que s'han realitzat sobre ells al llarg del temps. D'aquesta forma, el sistema és capaç de "recordar" les versions antigues de les dades, la qual cosa ens permet examinar l'històric de canvis o recuperar versions anteriors d'un fitxer, fins i tot encara que hagi estat esborrat.

Els sistemes de control de versions són àmpliament utilitzats en els projectes de desenvolupament de programari, per mantenir les versions del codi font. No obstant això, la seva aplicació no està limitada a aquesta activitat, sinó que permeten gestionar documents, imatges i fitxers de tot tipus. Aquests, tenen un seguit de vocabulari comú que els identifica:

**Repositori:** En primer lloc, els sistemes de control de versions utilitzen per al seu funcionament algun mecanisme d'emmagatzematge de les dades i la informació associada (metadades). Els sistemes de control de versions més populars funcionen amb un repositori centralitzat, és a dir, encara que permeten el treball col·laboratiu entre diversos llocs de treball, mantenen el repositori centralitzat en únic ordinador, estant accessible per a la resta d'equips a través de xarxa local o Internet.

Cada client del sistema, disposa de la seva pròpia còpia local de treball, que pot ser examinada i/o modificada a voluntat.

**Changeset:** Conjunt de canvis agrupats en diferents tipus de llistes. Un canvi és cadascuna de les modificacions realitzades en algun dels documents sota el control de versions. Molts sistemes permeten agrupar múltiples canvis en una sola operació d'escriptura o actualització del repositori.

**Nombre de versió:** Identificació d'una versió. Una revisió seria cadascuna de les versions disponibles en el repositori. Aquest terme s'utilitza referint-se tant a fitxers individuals com a conjunts de fitxers o fins i tot al repositori sencer.

Molts sistemes permeten definir etiquetes (*tags*) per referir-se a una revisió determinada de cert conjunt de fitxers. L'etiqueta permet assignar un nom fàcil de recordar o significatiu de manera col·lectiva a diversos fitxers alhora (per exemple, a tots els arxius corresponents a una mateixa versió publicada d'un programa).

**Branques:** Les branques permeten que a partir de cert punt, un mateix fitxer o conjunt de fitxers, sigui desenvolupat de dues formes diferents, mantenint les seves respectives revisions independentment.

Al procés d'actualització o escriptura dels fitxers baix control de versions, se li denomina integració. Aquesta paraula s'utilitza tant per indicar l'escriptura en el repositori dels canvis realitzats en la còpia de treball, com a l'actualització de la còpia local des del repositori per incloure els canvis realitzats per altres clients.

**Conflicte:** Quan en un procés d'integració, dos o més clients hagin realitzat canvis de manera independent sobre un mateix document. El sistema pot ser capaç de manejar alguns d'aquests conflictes, segons el cas, però en general, el procés de resolució (conciliació) dels mateixos sol requerir la intervenció de l'usuari.

### 5.4.1 - Git i avantatges

Git és un dels sistemes de control de versions més estesos actualment, la seva principal característica és que és distribuït. Neix de la mà de Lius Torvalds l'any 2005 per a gestionar codi font de Linux i poc a poc va anar guanyat adeptes en comunitats com GitHub o Bitbucket. El fet que sigui distribuït significa que no es necessita un servidor central, cada programador té una còpia completa del repositori en el seu equip i al fer un canvi es propaga a tots els nodes. Això permet en un gran rendiment en grans desenvolupaments, les cerques no són molt més eficaços el que suposa una gran rapidesa per detectar diferències entre arxius.

Git et dona poder per fer “*commit*” cada vegada que ho desitgis, inclòs quan no estiguis connectat en xarxa. A més, Git permet utilitzar fluxos de treball més flexibles que ajuden a treballar millor. I per suposat, al ser distribuït, qualsevol cerca que s'executi serà molt més eficient del que suposa una millora en temps de detecció de diferències entre arxius.

Als avantatges de Git cal sumar-hi que els repositoris i branques acostumen a ser molt més petits, que les branques són capaces de portar tot el seu historial sencer, que els formats d'arxiu dels repositoris són simples i la reparació és fàcil i la corrupció rara; el clonat de repositoris interacciona com a repositori de còpies de seguretat completes i proporciona una millor auditoria d'esdeveniments de ramificació (*branch*) i fusió (*merge*).

## 5.5 - Model-Vista-Controlador

El model-vista-controlador (MVC) és un patró d'arquitectura de software que separa les dades i la lògica de negoci d'una aplicació de la interfície d'usuari i el mòdul encarregat de gestionar els esdeveniments i les comunicacions. Per aconseguir-ho, aquest patró es nodreix de tres nuclis: el model, la vista i el controlador; és a dir, per una banda defineix components per a la representació de la informació i per un altre costat per a la interacció de l'usuari. Aquest patró es basa en la reutilització de codi i la separació de conceptes i responsabilitats.

En desenvolupament web aquest patró el trobem representat sobretot en els anomenats *frameworks*, eines de treball amb parts de codi preestablertes que estableixen la connexió entre les tres capes del patró.

### 5.5.1- Els components del MVC

De manera genèrica, els components del MVC es podrien definir com:

- Model: És la representació de la informació amb la que el sistema opera, per tant gestiona tots els accessos a aquesta: parlem d'escriptura, lectura i permisos d'accés. Envia a la vista aquesta part de la informació que a cada moment se li sol·licita, aquestes peticions arriben a través del controlador.
- Controlador: Respon a esdeveniments (normalment accions de l'usuari) i invoca peticions al model quan es produeix una sol·licitud sobre la informació, per tant es podria dir que funciona d'intermediari i manipula les dades segons com les requereixi la vista.
- Vista: presenta el model (informació i lògica de negoci) en un format adequat per a la interacció (normalment amb una interfície d'usuari) per tant, necessita que el model la retorni adequadament per a la seva sortida

### 5.5.2 - Els avantatges del MVC

1. Es pot dividir la lògica de negoci del disseny, fent el projecte més escalable.
2. Facilita l'ús de URL amigables, importants per a SEO (posicionament web), on la majoria de *frameworks* ho controlen.
3. Molts *frameworks* MVC ja inclouen llibreries de *Javascript* com *Jquery*, el que facilita validar formularis en el client i en el servidor.
4. Pots utilitzar abstracció de dades, com el Ruby on Rails, facilitant la realització de consultes a la base de dades
5. La majoria de *frameworks* controlen l'ús de la memòria caché, avui en dia és molt important per al posicionament web, ja que buscadors com Google donen prioritat a les web que tinguin menor temps de descàrrega.
6. En el cas de projectes on hi ha varis desenvolupadors, el seguir mètodes comuns de programació fa que el codi sigui més comprensible entre aquests, podent un continuar la feina de l'altre.
7. Els *frameworks* estan creats per facilitar el treball dels desenvolupadors, es troben tot tipus de classes predefinides i això influeix en la productivitat. Inicialment, com és lògic, hi ha una corba d'aprenentatge.
8. Poc a poc el desenvolupament web s'orienta al que venim anomenant "Desenvolupament web Àgil" i aquests *frameworks*, nodrits del patró MVC, en són els protagonistes.

9. Un *framework* MVC t'ajuda a controlar els recursos del servidor, evitant *bugs* que puguin repercutir en el rendiment, per exemple, moltes vegades ens no oblidem de tancar connexions a bases de dades, sobrecarregant el servidor.

## 5.6 - L'entorn del projecte

Com s'ha esmentat durant tota aquesta memòria, és important conèixer l'entorn on s'ha desenvolupat aquest projecte per conèixer més sobre la seva finalitat, requeriments i treball futur.

La idea de Boonobo surt d'un grup d'amics que sota el nom de la startup Octopi SL desenvolupen aplicacions a partir d'idees pròpies. La societat limitada és creada a partir de cinc socis el setembre del 2014 i a partir d'allà es realitzen varis *brainstormings* i s'elaboren plans d'estudis de viabilitat de cada una de les idees. Cada un dels socis és propietari d'un % de l'empresa i per tant de les idees i aplicatius que en aquesta es genera.

Un cop donada d'alta la societat, es crea un pacte de socis on s'informa de tots els possibles casos que es pot trobar en un futur la societat, i els seus socis, i com queden repetides les participacions. Aquest pas és important, ja que d'aquest document en depèn el futur de Boonobo i la part que en aquest projecte es desenvolupa.

A partir d'aquí, una acceleradora contacta amb mi i ens proposa desenvolupar l'eix de negoci i la idea de Boonobo; ens elaboren un pla d'acció, calendari administratiu, un pla de *marketing* i s'encarreguen de buscar empreses per a la prova pilot.

Sota la mentora d'aquesta acceleradora, i sempre amb l'acord dels cinc socis sobre qualsevol decisió que afectés directament al disseny de l'aplicatiu, s'ha desenvolupat aquest projecte. Les característiques del Sistema recomanador han afavorit al desenvolupament modular, de manera que funciona aliè a la implementació de la resta de l'aplicatiu.

L'aplicació web que en aquest projecte es presenta, com s'ha comentat anteriorment, està desenvolupada ad hoc per a l'abstracció de dades i mostra de resultats del sistema recomanador però, res té a veure amb l'aplicatiu final anomenat Boonobo i elaborat per Octopi.

### 5.6.1 - Startup

Una empresa *startup* o empresa incipient és un terme utilitzat actualment en el món empresarial que descriu l'acció d'engegar, emprendre o muntar un nou negoci i fa referència a idees de negocis que s'estan començant o estan en construcció, és a dir, són empreses

emergents recolzades en la tecnologia. Són idees que innoven el mercat i busquen fer dels processos complicats una via més fàcil; aquestes van enfocades a diferents temes i usos. Generalment són empreses associades a la innovació, al desenvolupament de tecnologies, al disseny web o al desenvolupament web: són empreses de capital-risc.

La *startup* d'on neix la idea d'aquest projecte és Boonobo, seria una startup del tipus incipient, és a dir, són les que generalment comencen amb una idea de negoci creatiu i el pas següent és agregar diferenciació a aquesta a través de la innovació per finalment, iniciar el negoci.

En aquest cas Boonobo és un projecte en actual desenvolupament per una altra empresa que l'acull, empresa formada pels mateixos socis que Boonobo, però que engloba altres *startups* i idees de negoci. Actualment l'aplicatiu està sent incubat per l'anomenada "acceleradora de *startups*" una institució que impulsa aquestes empreses a través d'un programa basat en convocatòries a curt termini de temps estimat. Aquests programes inclouen mentorització, formació intensiva, educació digital i tutorització. És tot un món que acompanya al projecte i que en forma part de la presa de decisions, inclòs algunes de les que han influenciat en aquest projecte. Un exemple en seria el disseny gràfic de l'aplicació, les seccions, els colors a utilitzar i les dades a recol·lectar.

## 6. Requisits del sistema

En aquesta part s'exposa el resultat final de l'etapa de recollida de requeriments que es va realitzar al principi del projecte. Per una banda, s'exposaran els requeriments funcionals, i per l'altre, els no funcionals. Els requeriments funcionals s'agrupen segons els actors que poden realitzar-los. Atesa la naturalesa d'aquest projecte, els requisits funcionals i no funcionals es dividiran segons si parlem de l'aplicatiu web o del sistema recomanador.

### 6.1 - Requisits de l'aplicació web: Boonobo

#### 6.1.1 - Requisits funcionals

Podem descriure un requisit funcional com una característica requerida del sistema que expressa una capacitat d'acció del mateix (una funcionalitat).

- Registre de l'usuari
- Iniciar i tancar sessió
- Visualitzar esdeveniments i filtrar-los segons categoria i espai temporal
- Visualitzar esdeveniments recomanats
- Visualitzar esdeveniments amb més assistents
- Visualitzar perfil i editar dades personals i preferències
- Visualitzar perfils d'usuaris
- Establir amistat amb altres usuaris
- Cercar i llistar usuaris de la plataforma
- Cercar i llistar usuaris amics
- Xatejar amb usuaris amics
- Visualitzar esdeveniment
- Assistir/No Assistir/M'agradaria/No Recomanar més sobre un esdeveniment
- Afegir/llegir comentaris en un esdeveniment
- Llegir la política de *cookies* i termes i condicions
- Visualitzar notificacions

#### 6.1.2 - Requisits no funcionals

Podem descriure un requisit no funcional com una característica requerida del sistema, del procés de desenvolupament, del servei prestat o de qualsevol altre aspecte del desenvolupament, que assenyala una restricció del mateix.

- Disposar d'un dispositiu amb connexió a *internet: smartphone, tablet* o ordinador
- Disponible des de qualsevol navegador
- El idioma de l'aplicatiu serà totalment en català (amb diferència de la informació de l'esdeveniment capturat).
- Coneixements bàsics de xarxes socials

## 6.2 - Requisits del sistema recomanador

Atesa la naturalesa d'aquesta part del projecte, no existeixen requeriments funcionals cap a l'usuari ja que és un procés que s'executa completament en el sistema. Si que te però, requeriments no funcionals per a la seva correcta utilització.

### 6.2.1 - Requeriments no funcionals

- Executar-se en una Màquina Virtual Java sota un JDE > 1.7
- Connexió a una base de dades MySQL amb les dades de connexió conegudes
- Disponible en qualsevol sistema operatiu que accepti execucions sobre la JVM
- Disponibilitat d'una consola o terminal de comandes per executar el sistema

## 7. Estudis i decisions

En aquest apartat es descriu el maquinari, les llibreries i/o el programari utilitzats durant el desenvolupament del projecte, i se'n justifiquen les eleccions. S'especifiquen també les eines utilitzades i el perquè de la tria d'aquestes eines.

### 7.1 - Llenguatges de programació

El projecte consta de dues parts: l'aplicatiu web i el *core* recomanador, per aquest motiu s'ha escollit el llenguatge que més s'ha adequat a les necessitats, requeriments i eficiència del sistema.

#### 7.1.1 - PHP & HTML5/CSS

**PHP**<sup>14</sup> és un llenguatge de programació d'ús general amb codi del costat del servidor originalment dissenyat pel desenvolupament web de contingut dinàmic. Principal motiu per la seva tria en aquest projecte: el seu objectiu de naixement és el món web.

Va ser un dels primers llenguatges de programació de servidor que podien incorporar directament el document HTML en comptes de cridar a un arxiu extern que processés les dades. És acceptat per la majoria de servidors web de la mateixa manera que tots els sistemes operatius i plataformes sense cost.

---

<sup>14</sup> És encara motiu de discussió i es desconeix l'origen de les sigles

Es considera un dels llenguatges més flexibles, més potents i d'alt rendiment coneguts fins el dia d'avui, fet que ha atret el interès de sectors amb molta potència com *Facebook*. Forma part del software lliure publicat sota llicència PHP que és compatible amb la Llicència Pública General de GNU després de les restriccions de l'ús del terme *PHP*. Han estat la gran enumeració d'avantatges que té aquest llenguatge el que ha decidit que s'utilitzés pel desenvolupament d'aquest projecte:

- Com hem comentat és un llenguatge lliure i obert.
- Té una corba d'aprenentatge molt baixa
- Els entorns de desenvolupament són fàcils de configurar com veurem en el *framework* escollit
- Els paquets del PHP són autoinstal·lables pel que no ens hem de preocupar de la seva gestió.
- Consta d'un fàcil accés a la BdD a diferència d'altres llenguatges com és el Java
- És multi plataforma: acceptat per tots els sistemes operatius afavoreix la seva abstracció i instal·lació/execució en qualsevol servidor
- Estable, flexible i ràpid són les tres característiques que el diferencien d'altres llenguatges

**HTML5** (*Hypertext Markup language* versió 5) és la cinquena revisió important del llenguatge bàsic de *World Wide Web*, HTML. Aquesta versió especifica dues variants de la sintaxis per a HTML: una clàssica denominada HTML5 i una variant com XHTML5 que es serveix de sintaxis XML. A causa no ser reconegut en velles versions de navegadors per les seves noves etiquetes, es recomana a l'usuari navegar per aquestes pàgines amb el navegador més nou per poder gaudir de tot el potencial; això no significa que es perdi informació de per mig. HTML5 estableix una sèrie de nous elements i atributs que reflecteixen l'ús típic de les webs modernes com per exemple *<nav>* i *<footer>*. Aquestes millores no només ajuden a la comprensió i a una millor presentació del contingut web sinó a l'ús modular de la informació que és el que es busca sempre en disseny de software.

Segueix amb la estandardització marcada per a W3C i estableix més perfils que la seva primera versió coneguda simplement com a HTML. Es considera el llenguatge web més important sent la seva invenció clau per a l'aparició, desenvolupament i expansió de la WWW. No és d'estranyar doncs que totes les apostes fetes sobre HTML5 anessin a millorar aquests orígens de la primera versió.

En el llenguatge HTML existeixen diferències força notables amb altres llenguatges, per exemple, per inserir contingut com un text, imatges, vídeo.. externs a la pàgina, aquest no s'incrusta directament sinó que es fa una referència a la ubicació d'aquest element.

**La fulla d'estils en cascada o CSS** (*Cascading Style Sheets*) és un llenguatge utilitzat per a definir i crear la presentació d'un document estructurat i escrit en HTML. La idea principal que es troba en aquesta abstracció és separar l'estructura d'un document de la seva presentació i estils. També ajuda a la reducció de codi i al seu re aprofitament.

Els avantatges i motius pels quals s'ha escollit HTML5 (i per tant també CSS) són:

- És natiu i per tant independent a *plugins* i navegadors de tercers
- Permet classificar i ordenar els diferents nivells de codi. Proporciona meta dates que afavoreixen al posicionament SEO
- Codi simple que permet fer pàgines simples i de ràpida càrrega
- Més compatibilitat amb navegadors de dispositius mòbils, adient per les característiques del nostre projecte que requereix més mobilitat
- Permet disseny *responsive*<sup>15</sup> 2001%

### 7.1.2 - Java

Java és un llenguatge de programació de propòsit general, concurrent i orientat a objectes que va ser dissenyat específicament per tenir tan poques dependències d'implementació com sigui possible. La seva filosofia WORE<sup>16</sup> permet que un cop compilat, es pugui executar en qualsevol entorn que disposi d'una màquina virtual java. Gràcies a un dels seus objectius com la facilitat d'ús i execució, aquest llenguatge que té com a paradigma la programació orientada a objectes, és un dels llenguatges de programació més utilitzats.

No és casualitat que gran programes per a mineria de dades, aprenentatge automàtic i anàlisi de dades estiguin escrites en Java. Els grans aplicatius com *Apache Mahout*, *Apache Hadoop* i *Tomcat* són exemples de com la filosofia WORE és una realitat.

Imaginem per un instant que podem executar el mateix arxiu en diferents sistemes operatius sense haver d'aplicar cap canvi.

Un altra avantatge és la integració amb les aplicacions abans esmentades, sense necessitat d'accedir-hi via API, sinó senzillament afegint els mòduls i llibreries, ens facilita moltíssim la feina.

*ElasticSearch*, servidor de cerca que en un bon inici s'havia d'utilitzar en aquest projecte, també està implementat amb Java i ofereix molts possibilitats d'integració amb programes

---

<sup>15</sup>Filosofia de disseny web que permet adaptar la web al dispositiu que l'estigui visualitzant sense perdre l'estructura desitjada.

<sup>16</sup>*Write once, run everywhere*

escrits sota el mateix llenguatge. Va ser un altre dels motius inicials pels que es va triar programar el SR en Java.

Un dels altres motius per haver escollit Java és la part analítica que ens proporciona sobre les dades; si bé en un bon inici es volia utilitzar R, després veurem que utilitzem una altra llibreria Java, aquest és de fàcil integració en una aplicació escrita en *java* gràcies als *plugins* que se'ns proporciona. Tradicionalment ja és un llenguatge que utilitzem per a tots tipus d'anàlisi de dades, la gran quantitat de llibreries existents són fruits de la filosofia Java i la orientació a objectes. No és d'estranyar doncs que en un sistema recomanador, on un dels principals objectius és la tracta de dades, també s'utilitzi Java.

## 7.2 - Framework CakePHP

*CakePHP* és un marc de desenvolupament ràpid per a PHP, lliure i de codi obert. Es tracta d'una estructura que serveix com a base als programadors perquè aquests puguin crear aplicacions web. El seu principal objectiu és que es pugui treballar de forma estructurada i ràpida, sense pèrdua de flexibilitat. Només amb la còpia d'un projecte buit original, es pot començar a escriure la lògica de l'aplicació web.

Com a característiques tècniques cal destacar que és de llicència flexible, compatible amb la última versió de PHP, integració ràpida amb les bases de dades gràcies a CRUD, té un auto generador de codi, té validació integrada, tot una llibreria de *helpers* escrits en PHP que funcionen de plantilles web, components d'*email-cookies-seguretat* i sessió, neteja automàtica de dades i *cache* flexible. D'aquestes característiques les més importants és la gestió de *cookies*, seguretat i sessió: realment dels *frameworks* coneguts és el que proporciona una gestió més simple i fàcil d'entendre.

Necessita córrer sobre un servidor Apache i això és un punt més a favor, ja que redueix l'aprenentatge previ per aquest projecte (al tenir coneixements bàsics d'Apache).

Altres motius per la tria d'aquest *framework* és la facilitat de reutilització de codi que proporciona i el patró MVC que implementa; com havíem descrit abans, era el més adient per la realització del nostre aplicatiu web i són pocs els *frameworks* PHP força complets que l'implementen amb moltes facilitats per al desenvolupador.

A continuació s'adjunta una petita comparativa amb un altre *framework* com és Symfony. Actualment Symfony és el més utilitzat, tot i que Cake es va guanyar el primer lloc en la seva versió 2.1, s'afegeix una petita comparativa d'algunes característiques dels dos marcs:

Característiques	CakePHP	Symfony
Operacions CRUD (Create, Retrieve, Update i Delete) associades al patró Active Record)	Sí	Sí
Generació de URLs amigables	Sí	Sí
Ús de plantilles en PHP.	Sí	Sí
Maneig de privilegis d'accés a seccions d l'aplicació (Acces Control List).	Sí	No
Emmagatzematge a la caché de la configuració de l'aplicació.	No	Sí
Eines de prova i depuració	No	Sí
Interfície de línia de comandes per a la creació i manteniment d'aplicacions.	Sí	Sí
Emmagatzematge en caché de les vistes.	Sí	No
Emmagatzematge de logs de funcionament del framework.	No	Sí
Implementació pròpia de crides Asynchronous Javascript i XML (AJAX)	Sí	Sí

Si bé Symfony supera per moltes característiques a CakePHP i també implementa MVC, no ha estat escollit com a *framework* de desenvolupament per l'augment de temps que suposaria en el projecte per el seu aprenentatge. Com recordem, l'aplicatiu web no s'utilitzarà en un treball futur, per tant no valia la pena invertir-hi més temps del necessari.

### 7.3 - Servidor Amazon Web Services

Amazon Web Services (AWS) és una col·lecció de serveis de computació en el núvol (també anomenats serveis web) que en conjunt formen una plataforma de computació en el núvol, o ferides a través d'Internet per Amazon.com. És utilitzat en aplicacions populars com Dropbox, Hootsuit i Forsquare, que en el seu dia es van desfer dels servidors físics per llogar-ne en el núvol. És una de les ofertes internacionals més importants de la computació en el núvol i competeix amb serveis com Microsoft Azure i Google Cloud Platform: és però el primer i de moment, el més contractat entre aquests.

Abans d'explicar del perquè d'Amazon en comptes d'algun dels seus competidor, és important tenir clar perquè és millor contractar un servidor al núvol que a més et proporciona motors a part del simple emmagatzematge.

- **Facilitat d'ús:** AWS està dissenyat per permetre a tot tipus d'usuaris i empreses emmagatzemar la seva informació i desplegar-hi les seves aplicacions. Està accessible des de consola o des d'una excel·lent API documentada.
- **Flexibilitat:** AWS ens permet seleccionar el sistema operatiu, el llenguatge de programació, la plataforma web, la base de dades, així com la resta de serveis que necessitem de manera auto instal·lable. Això significa oblidar-nos en gran part del procés d'instal·lació i configuració.
- **Preu:** AWS només ens farà pagar per realment allò que gastem: depenem del tràfic, de l'espai que utilitzem i la velocitat de descàrrega. Sense contractes ni compromisos per avançat. No només ens permet portar un control de les despeses i fer pautes de despeses sinó que a més, ens permet fer una prova d'un any completament gratis amb una màquina senzilla.
- **Fiabilitat:** Disposarem d'una infraestructura informàtica global, escalable, fiable, segura, columna vertebral de la famosa botiga Amazon.com, botiga valorada en milions de milions de dòlars que porta en el mercat des de fa més de 10 anys.
- **Escalabilitat i alt rendiment:** amb les eines de AWS, Auto Scaling i Elastic Load Balancing, la seva aplicació podrà ampliar-se o reduir-se segons demanda. Disposa de milers de servidors en varies centenars de regions de franges horàries diferents.
- **Seguretat:** AWS aplica un enfocament integral en la protecció i reforç de la seva infraestructura que contempla mesures físiques, operatives i de *software*. Proporciona perfils configurables de seguretat i patrons HTTP i HTTPS configurables.

Els dos grans líders actuals de la *cloud-computing* són Microsoft Azure i AWS que junts sumen el 70% d'aquest tipus de contractacions. Seguidament se n'adjunta una comparativa entre aquests que reforça el perquè s'ha escollit AWS:

Característiques	M. Azure	AWS
Opcions de computació (instàncies, regions i zones)	4 instàncies diferents, sí que admet regions però no cloud per zones horàries	7 instàncies diferents, sí que admet regions i té cloud per zones horàries
NoSQL i Big Data	DynamoDB, Kinesis EMR i Redshift	Windows Azure table
Virtual Network i configuració de DNS	VNet sense configuració de DNS	VPN i configuració completa del DNS
Característiques del servidor	Màquina virtual que cal instal·lar sistema operatiu (Windows o Linux redHat) entre 25 tipus de servidor	Core computeEC2, conjunt de paquets instal·lables entre tots els sistemes operatius; entre 53 tipus de servidor
Quantitat de temps anual de fallada del servidor en mitjana	5 hores	13 minuts
Facilitat de complementació amb altres productes	Sincronització amb tots els productes Microsoft	Ofereix plugins per a connexió amb tot tipus de softwares lliures/privats

Preu per billions d'operacions en Benchmarks	0,5\$ pagament per minut sense rebaixa	1,5\$ però amb versió d'un any gratuïta, pagament per hora amb rebaixes mensuals
Temps de prova	30 dies	1 any
Marketplace	707 aplicacions	2400 aplicacions
Seguretat	25 certificacions	20 certificacions

Entre dos serveis bastant competitius, es tria per Amazon al tenir un *marketplace* més ampli per fer proves d'analítica de dades, un període de prova més alt i una quantitat d'usuaris en la xarxa, que facilita la resolució de dubtes (tutorials i FAQs).

Podria haver agafat diferents mòduls AWS com emmagatzematge web i un mòdul SQL però s'ha optat per la decisió d'agafar un mòdul de servidor EC2 que permet la instal·lació d'un Ubuntu amb un servidor LAMP, servidor utilitzat amb anterioritat, de fàcil configuració i ens evitàvem estudiar la connexió entre mòduls AWS.

### 7.3.1 - LAMP

LAMP és l'acrònim utilitzat per a descriure un sistema d'infraestructura d'Internet que utilitza les següents eines:

- Linux: el sistema operatiu.
- Apache: el servidor web
- MySQL/MariaDB: el gestor de bases de dades
- PHP/Perl o Python: llenguatges de programació

La combinació d'aquestes tecnologies és utilitzada principalment per a definir la infraestructura d'un servidor web, utilitzant un paradigma de programació per al desenvolupament. Tot i que l'origen d'aquests programes de codi obert no han estat específicament dissenyats per a treballar entre sí, la combinació es va popularitzar gràcies al seu baix cost d'adquisició i ubiquïtat dels seus components (ja que venen preinstal·lats en la majoria de les distribucions Linux). Quan són combinats, representen un conjunt de solucions que suporten servidors d'aplicacions.

## 7.4 - Base de dades MySQL

MySQL és un sistema de gestió de base de dades relacional, multi fil i multi usuari.

A diferència de les altres tecnologies descrites en aquest apartat, la base de dades venia imposada abans d'iniciar el projecte. Com que canvia molt quina base de dades utilitzem per la tria del servidor i del sistema recomanador, aquest es va establir com a requisit a la descripció del projecte.

MySQL és la base de dades utilitzada per l'aplicatiu original de Boonobo i com que utilitzem una còpia de la base de dades, el sistema recomanador s'ha de basar en aquesta.

Per tant, la única manera de justificar-ho és com a requisit.

## 7.5 - Git

Abans hem parlat d'aquest control de versions, que a part de ser el més conegut, és el que s'estudia a la carrera i el que s'utilitza en l'aplicatiu original Boonobo.

## 7.6 - Repositori Bitbucket

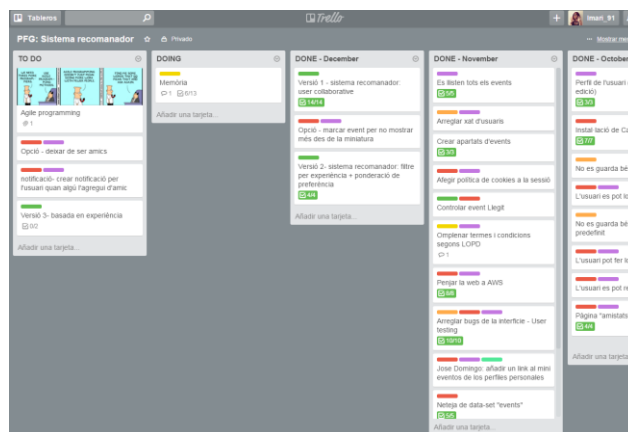
Bitbucket és un servei d'allotjament basat en web, per als projectes que utilitzen el sistema de control de versions Mercurial i Git. Bitbucket ofereix plans comercials i gratuïts amb limitacions diferents. És un repositori que ofereix serveis de comptes gratuïtes i comptes de pagament segons el nombre d'usuaris accessibles al repositori.

Les dues grans empreses de repositoris per a Git són GitHub i Bitbucket i s'afegeix una comparativa que ajuda a fer-se una idea dels motius pels quals s'ha escollit el segon:

Característiques	GitHub	Bitbucket
Control de versions acceptats	Git	Git i Mercurial
Número de projectes/repositoris per compte gratuïta	Il·limitats però tots OpenSource amb col·laboradors il·limitats	Repositoris i projectes il·limitats amb 5 col·laboradors tant privats com públics
Número de projectes/repositoris pagant	7 repositoris privats i tancats per 7€/mes	Augment col·laboradors il·limitats per 10€/mes
Pull request	Sí	Sí
Integració amb Jira	No	Sí
Característiques de xarxa social	Sí	No

Precisament perquè no ens interessa que el codi del sistema recomanador i de l'aplicatiu siguin públics i altres desenvolupadors hi puguin col·laborar, s'ha triat Bitbucket (a més ja es disposava d'un compte en aquest).

## II-lustració 9 Board trello del PFG



## 7.7 - Trello board

Trello és una aplicació web (està en el núvol) per fer tasques dins de llistes (descrivint-la d'una manera molt bàsica). El que la fa única és la seva versatilitat. A cada element d'una llista se li pot agregar de tot: altres llistes, imatges, vídeos, documents, etc. A més, és extremadament potent per a ús col·laboratiu. Poden agregar-se qualsevol nombre d'usuaris, assignar tasques, posar-los data límit, etc. Es basa en el mètode Kanban per a gestió de projectes, amb targetes que viatgen per diferents llistes en funció del seu estat: Així, solem tenir una llista de coses per fer (to do, o pendants), que s'estan fent (doing, o en procés) o fetes (doni, o acabades).<sup>17</sup>

És ideal per a la gestió de projectes i *backlogs* amb metodologia àgil: permet el control del projecte, l'elaboració de *sprints* a través de *due-dates* (dates de venciment) i la gestió gràfica de les tasques i les subtasques tal com s'especifiquen en les *features* en el *backlog*. S'ha escollit per controlar de manera dinàmica les tasques i poder-ne incorporar de noves després del feedback dels usuaris.

## 7.8 - IDE Eclipse

Eclipse és un programa informàtic, anomenat IDE o Entorn de Desenvolupament Integrat, compostat d'un conjunt d'eines de programació de codi obert multiplataforma per a desenvolupar el que anomenen com "Aplicacions de Client Enriquit" oposat a aquelles que es basen en navegadors. Pròpiament és un IDE de Java però, proporciona molts *pluguins* per treballar amb diferents llenguatges i servidors interns.

Ja que és el IDE amb el que més havia treballat i ja estava configurat per a treballar amb Java i Eclipse, i també perquè proporciona una bona interfície per a treballar directament amb Git, ha estat l'IDE escollit per a treballar durant tot el projecte.

---

<sup>17</sup> "How to Organize Your Entire Life with Trello - Lifehacker." 2015. 14 Jul. 2015 <[www.lifehacker.com](http://www.lifehacker.com)>

## 7.9 - Tècniques de recomanació

Les tècniques de recomanació escollides han estat dues: per a la base un sistema col·laboratiu i per a la millora de la sortida, un filtratge basat en el contigut.

El motiu per escollir aquests dos ha estat la recerca prèvia per a la documentació del projecte. Els principals portals basats en sistemes recomanadors es basen en patrons col·laboratius donada l'evolució cada dia més social que s'està donant a Internet. La creació de xarxes i societats virtuals ha fet evolucionar els SR per adaptar-se a les mateixes situacions que es troben a la vida real: Les decisions condicionades per la resposta majoritària i els consells proporcionats per aquells que conformen el nostre entorn.

Cal però, tenir en compte el propi perfil de l'usuari, per tant sempre s'ha de puntuar més o menys alt aquelles recomanacions que concordin amb els gustos de l'usuari o, que pel contrari, siguin contràries a aquestes.

La majoria d'algorismes de filtratge col·laboratiu que existeixen a dia d'avui, inclosos algorismes purs recomanadors, treballen generant primer les prediccions basant-se en les preferències de l'usuari i després produir recomanacions basades en el *ranking* de puntuacions proporcionades pel veïnat dins d'aquestes preferències. Si bé és cert que aquest treball no deixa de ser un treball d'investigació obre un món que no coneix de regles concretes: ens hem basat en el mateix model comú però s'ha invertit.

Hi ha dos maneres d'encarar aquest projecte: des del punt de vista de l'Usuari i des de punt de vista de l'Ítem. A l'aplicar després un filtre basat en preferències d'usuari i no en "*Morelike this*" esdeveniments, s'ha optat pel primer model.

Els Sistemes Recomanadors Col·laboratius basats en usuaris són de senzilla implementació algorísmica amb les premisses com a nucli d'un filtratge col·laboratiu: buscar a altres usuaris on part dels seus ràtings passats siguin similars als de l'usuari que actualment estem analitzant i predir què li agradarà a l'usuari. Es recomana enfocar-ho d'aquest model quan tenim molts més ítems que usuaris en la nostra base de dades, i agafar el segon model quan es doni la situació contrària. Ambdós mètodes necessiten una mínima de computació *offline* i la despesa depenent de com de gran sigui la demanda computacional quan la necessiti el SR (la pre-computació del model basat en l'ítem implementa un temps d'execució de *queries*<sup>18</sup> més alt). El sistema basat en l'usuari sembla que proporciona una millor serindipia<sup>19</sup> en les seves recomanacions; en el cas del projecte MovieLens, això es va traduir en una millor satisfacció de l'usuari.

<sup>18</sup> Parlarem de queries com a query SQL que es fa a la base de dades del projecte

<sup>19</sup> Efecte que es dona quan accidentalment es fa una troballa quan es buscava quelcom gens relacionat.

Quins són els problemes del sistema basat en contingut, que necessitem construir un híbrid per a millorar-lo?

Necessitem saber-ho tot absolutament de l'ítem que estem analitzant i en el cas d'esdeveniments capturats de Internet no sempre tenim la mateixa informació de tots, cosa que dificultaria aquest model pur. Triar quines característiques els identifiquen millor, seria tot un repte impossible de resoldre 100%. A més, com solucionem el problema del *cold-start* o inici en fred de l'usuari? Quan aquest accedeix a l'aplicació no tenim cap mena de preferències d'aquest i igualment hem de recomanar-li quelcom. A més hi ha un factor que es dóna en el procés de la cerca de l'ítem que satisfà la nostra necessitat i és el de serindipia: hauríem de donar la possibilitat de trobar coses que no estiguin relacionades amb el que busquem.

Tots aquests problemes queden resolts amb el model col·laboratiu, és per aquests motius que s'ha triat construir aquest model híbrid per aquest projecte.

## 7.10 - Llibreria Apache Maths

Common Maths és una llibreria lleugera i autònoma que agrega components estadístics que aborden els problemes més comuns que no estan disponibles en el llenguatge de programació Java o Commons Lang<sup>20</sup>. Les característiques que l'identifiquen són:

- Els casos d'ús d'aplicacions del món real determinen la prioritat del desenvolupament.
- Aquest paquet remarca petits components, fàcilment integrats en lloc de grans biblioteques amb dependències i configuracions complexes.
- Tots els algorismes estan totalment documentats i segueixen les millors pràctiques de programació generalitzades
- En situacions en què hi ha múltiples algorismes estàndard, s'utilitza un patró d'estratègia per suportar múltiples implementacions.
- No hi ha dependències externes més enllà dels components dels Common i la plataforma central Java (com a mínim amb Java 5)

La seva utilització és tan fàcil com afegir la llibreria a les dependències del projecte descarregant-ne el .jar que es proporciona a la web d'Apache. Respon a la resolució de més de 13 famílies de problemes matemàtics diferents com són muntatge de línies a través de

---

<sup>20</sup>És un paquet amb classes d'utilitat Java per aquelles que es troben en l'herència del `java.lang`'s.

regressió lineal, interpolació d'una corba, resoldre equacions, generació de dades i conjunt de mostres... i es classifiquen en 16 subpaquets basats en funcionalitats.

Ateses les característiques dels problemes que havíem de resoldre, els dos paquets utilitzats són el d'estadística i en d'anàlisi de dades.

La gran decisió que s'ha pres sobre quina llibreria a utilitzar ha marcat en gran part el desenvolupament del sistema recomanador. En un principi es volia treballar amb el servidor R<sup>21</sup> i l'API que proporciona per utilitzar tots els serveis estadístics que té. S'havia trobat el connector JRI, una llibreria que serveix de pont entre el servidor R i qualsevol projecte Java però aquí se'ns presenta el primer problema: requeria d'una configuració diferent per a Windows que per Unix (variables d'entorn, configuració de la màquina Virtual Java) i això duplicava la feina d'aprenentatge ja que, recordem el servidor d'Amazon és un Ubuntu.

Tot i així es va decidir tirar endavant i aprendre el llenguatge de rJava, la interfície per Java per accedir a totes les operacions disponibles en el servidor R i aquí va aparèixer el segon problema: la meua versió d'Eclipse no reconeixia la interfície rJava i buscant l'error que donava vaig trobar un *bug* report al projecte JRI amb les mateixes condicions amb les que em trobava jo. Sense poder provar la interfície, haver d'aprendre un seguit de crides noves amb Java (gens simples) i havent de tenir el servidor R sempre engegat (amb una configuració que a Unix seria diferent i a més el servidor Amazon gratuït tampoc hagués suportat la quantitat de crides internes) es va decidir estudiar altres solucions.

Quina va ser la meua sorpresa de que Apache tenia una llibreria específica que resolvia les meves necessitats. En un projecte de desenvolupament com aquest, el temps és or. I si la mateixa feina la pots fer per dos camins, tria el més curt.

## 7.11 - ElasticSearch

En un principi s'havia d'utilitzar aquest motor de cerca per indexar els esdeveniments i no només poder implementar un cercador per paraules més avançat i amb un alt grau de rendiment; la indexació ens facilitaria poder utilitzar la propietat "*More like this*" i a través de quèries amb unes característiques concretes dels esdeveniments (títol, descripció, categoria..) trobar esdeveniments semblants entre ells. I no és que ara no faci servei, la realitat és que serà utilitzada en el projecte original i no en el sistema recomanador. Per tant de poc serveix implementar una eina en aquest projecte que no hi va estrictament relacionat.

---

<sup>21</sup> R és un llenguatge i entorn de programació per a l'anàlisi estadístic i gràfic.

Quin va ser el motiu d'incloure'l a la descripció del projecte? Com s'ha comentat anteriorment, el problema dels sistemes recomanadors es pot enfocar de maneres molt diferents i des d'un inici es volia utilitzar la tècnica “*More Like This*” per mostrar esdeveniments semblants a l'usuari. En el moment en que molts esdeveniments els hi manquen molts camps que a la base de dades estan a nul, només ens podem guiar pel títol i la descripció. Tenint en compte que aquests ja són comparats prèviament a aquest projecte per realitzar la classificació de categoria, no val la pena tornar a fer les comparacions.

Després, cal afegir que a l'estar tot el sistema recomanador enfocat des del punt de vista d'Usuari, són pocs els motius per continuar utilitzant aquesta eina.

## 7.12 - **Dataset: Esdeveniments i Usuaris**

Els dos *datasets* necessaris per dur a terme el Sistema Recomanador són Esdeveniments i Usuaris. Els dos han estat recollits amb dades reals la diferència que els Esdeveniments era un conjunt capturat de Facebook anterior a aquest projecte i Usuaris ha estat recollit durant el desenvolupament i proves.

Les esdeveniments venen donats com a *dataset* d'entrada, no els podem modificar i hem de tractar-los com a reals. Molts d'ells tenien dates antiquades o passades i l'únic que s'ha fet és sumar els mateixos mesos per a tots per recompensar el temps passat. Tenim un total de 7450 esdeveniments classificats, també amb anterioritat a aquets projecte, entre 14 categories diferents. Les categories també venen donades amb anterioritat com explicarem més endavant en el model de dades.

Els usuaris s'han recollit a través d'un procés de col·laboració, es va enviar un document amb els passos a seguir a tots aquells que van mostrar interès en el projecte. És així com s'han recopilat un total de 72 perfils diferents d'usuaris amb les seves preferències i activitat a la plataforma.

## 8. Anàlisi i disseny del sistema

A continuació es farà un anàlisi dels requeriments funcionals mitjançant un seguit de diagrames de casos d'ús i les seves fitxes corresponents, i es presentarà un disseny que dona una solució, amb el diagrama de classes.

### 8.1. Diagrames de casos d'ús

Considerant les característiques d'aquest projecte, que es divideix en dos blocs, no existeixen requeriments funcionals a la part del sistema recomanador, no almenys des de la visió de l'usuari per tant només s'especificaran aquells que provenen de la web. Els diferents actors que interactuen amb el sistema són:

1. Usuari no autenticat.
2. Usuari autenticat.

A partir dels requeriments i dels diferents actors s'ha definit el diagrama de casos d'ús:



II-lustració 10 - Diagrama casos d'ús

## 8.2 - Fitxes de casos d'ús

Cas d'ús	Llegir termes i condicions
Descripció	L'usuari pot llegir els termes i condicions de la pàgina
Actor	Usuari no autenticat
Precondició	L'usuari ha d'haver anat a la pàgina de registre
Flux principal	<ol style="list-style-type: none"> <li>1.Usuari indica que vol llegir els termes i condicions</li> <li>2. Se li mostren els termes i condicions               <ol style="list-style-type: none"> <li>a. L'usuari prem "accepto"</li> <li>b. L'usuari tanca la finestra</li> </ol> </li> </ol>
Fluxos alternatius	3. En cas d'error, es mostra un missatge indicant-lo
Postcondició	L'usuari ha acceptat/tancat els termes i condicions

Cas d'ús	Registrar-se
Descripció	L'usuari pot registrar-se per accedir a la web
Actor	Usuari no autenticat
Precondició	L'usuari ha d'haver anat a la pàgina de registre
Flux principal	<ol style="list-style-type: none"> <li>1.Usuari omplena el formulari correctament</li> <li>2. Tria l'opció de "Registre"</li> <li>3. L'usuari és redirigit a la pàgina principal</li> </ol>
Fluxos alternatius	4. En cas d'error, es mostra quins camps ho provoquen
Postcondició	L'usuari s'ha registrat correctament

Cas d'ús	Login
Descripció	L'usuari pot loguejar-se per accedir al portal web
Actor	Usuari no autenticat
Precondició	L'usuari ha d'estar registrat,Usuari està a la pàgina principal
Flux principal	<ol style="list-style-type: none"> <li>1.Usuari omple el formulari d'autenticació</li> <li>2.Usuari prem el botó "Login"</li> <li>3. El sistema comprova les dades <ol style="list-style-type: none"> <li>3.1 Si les dades són correctes, s'inicia sessió i s'ens redirigeix a la pàgina de mostrar recomanats</li> </ol> </li> </ol>
Fluxos alternatius	4. Si hi ha error, es mostra per pantalla
Postcondició	L'usuari ha quedat loguejat i a la pàgina d'inici de la plataforma

Cas d'ús	Acceptar política de cookies
Descripció	L'usuari pot acceptar la política de cookies
Actor	Usuari no autenticat /autenticat
Precondició	Haver entrat a la pàgina i no haver-ho acceptat amb anterioritat
Flux principal	<ol style="list-style-type: none"> <li>1. L'Usuari prem "ok"</li> <li>2. El sistema crea una cookie d'acceptació</li> <li>3. Es tanca el pop-up: la política ha estat acceptada</li> </ol>
Fluxos alternatius	4 . En cas d'error, no es tancaria ni es crearia la cookie
Postcondició	S'ha acceptat la política de cookies

Cas d'ús	Tancar sessió
Descripció	L'usuari pot tancar sessió de l'aplicació
Actor	Usuari autenticat

Precondició	L'usuari ha d'estar loguejat
Flux principal	<ol style="list-style-type: none"> <li>1. L'usuari prem el botó situat a totes les pantalles "Tancar sessió"</li> <li>2. El sistema tanca la sessió</li> <li>3. L'usuari és redirigit a la pàgina de login</li> </ol>
Fluxos alternatius	4. En cas d'error, es retorna a la pàgina on estàvem amb un missatge
Postcondició	S'ha tancat la sessió de l'Usuari

Cas d'ús	Visualitzar esdeveniments
Descripció	L'usuari pot veure una llista de tots els esdeveniments
Actor	Usuari autènticat
Precondició	L'usuari ha d'estar autènticat
Flux principal	<ol style="list-style-type: none"> <li>1. Usuari va a l'opció del menú esquerre a "Events" i després "Tots els events"</li> <li>2. El sistema llista tots els esdeveniments en qualsevol temps de qualsevol categoria</li> <li>3. L'usuari escull un filtre per categoria <ol style="list-style-type: none"> <li>a. El sistema mostra només per el temps especificat els esdeveniments d'aquella categoria</li> </ol> </li> <li>4. L'usuari escull un filtre de temps <ol style="list-style-type: none"> <li>a. El sistema mostra només per el filtre de categoria especificat, els esdeveniments en aquell període de temps</li> </ol> </li> </ol>
Fluxos alternatius	5. En cas d'error, la llista d'esdeveniments es manté intacta
Postcondició	L'Usuari visualitza una llista d'esdeveniments

Cas d'ús	Visualitar esdeveniments recomanats
Descripció	L'usuari pot visualitzar els esdeveniments que se li recomanen
Actor	Usuari autènticat
Precondició	L'usuari ha d'estar autènticat
Flux principal	<ol style="list-style-type: none"> <li>1. L'Usuari va a l'opció del menú esquerre "Inici" o "Events" i després a "Events recomanats" o simplement és la vista de la pàgina principal</li> <li>2. El sistema llista tots els esdeveniments que tenen la propietat "recomanat" per a aquell usuari en qualsevol temps</li> <li>3. L'usuari escull un filtre per categoria <ol style="list-style-type: none"> <li>a. El sistema mostra només per el temps especificat els esdeveniments recomanats d'aquella categoria</li> </ol> </li> <li>4. L'usuari escull un filtre de temps <ol style="list-style-type: none"> <li>a. El sistema mostra només per el filtre de categoria especificat, els esdeveniments recomanats en aquell període de temps</li> </ol> </li> </ol>
Fluxos alternatius	5. En cas d'error, la llista d'esdeveniments es manté intacta
Postcondició	L'Usuari visualitza una llista d'esdeveniments recomanats

Cas d'ús	Afegir amistat
Descripció	L'usuari pot visualitzar el perfil de l'usuari b i afegir-lo com amistat (si no ho són)
Actor	Usuari autènticat
Precondició	L'usuari ha d'estar autènticat Hem d'haver llistat el perfil de l'usuari b

Flux principal	<ol style="list-style-type: none"> <li>1. El sistema mostra a l'Usuari el perfil de l'usuari b</li> <li>2. L'Usuari visualitza el perfil de l'usuari b</li> <li>3. Si l'Usuari i l'usuari b no eren amics <ol style="list-style-type: none"> <li>a. El sistema mostra el botó "afegir amistat"</li> <li>b. L'Usuari prem el botó</li> <li>c. El sistema crea la relació d'amistat</li> <li>d. L'Usuari i l'usuari b ja són amics</li> <li>e. L'Usuari pot visualitzar el perfil complet de l'Usuari</li> </ol> </li> <li>4. Si l'Usuari i l'usuari b ja eren amics <ol style="list-style-type: none"> <li>a. El sistema mostra una part més de perfil de l'usuari b</li> <li>b. El sistema mostra el terme "Amics"</li> </ol> </li> </ol>
Fluxos alternatius	5. En cas d'error, el sistema ens informa amb un missatge
Postcondició	L'Usuari afegeix l'usuari b com amic / L'Usuari i usuari b ja éren amics

Cas d'ús	Editar perfil
Descripció	L'usuari pot editar el seu perfil
Actor	Usuari autènticat
Precondició	L'usuari ha d'estar autènticat

Flux principal	<ol style="list-style-type: none"> <li>1. L'usuari prem l'opció "perfil" del desplegable de la barra superior de menú</li> <li>2. El sistema mostra el perfil de l'Usuari</li> <li>3. l'Usuari pot editar el seu perfil: editant el formulari, escollint un nou avatar, escollint preferències de categories, visualitzant els esdeveniments assistits i veure la seva ubicació (per defecte Girona).</li> <li>4. Si l'usuari prem "seleccionar avatar" <ol style="list-style-type: none"> <li>a. El sistema mostra un desplegable amb tots els avatars disponibles</li> <li>b. Si l'Usuari prem un avatar <ol style="list-style-type: none"> <li>i. Es tanca la finestra i s'especifica com a nou avatar</li> </ol> </li> <li>c. Si l'Usuari tanca la finestra <ol style="list-style-type: none"> <li>i. L'avatar de l'usuari no és substituït</li> </ol> </li> </ol> </li> <li>5. Si l'Usuari prem "Guardar canvis" <ol style="list-style-type: none"> <li>a. El sistema guarda els canvis</li> <li>b. L'Usuari és notificat de que els canvis s'han dut a terme i es torna a redirigir al perfil</li> </ol> </li> <li>6. Si l'usuari marxa sense guardar <ol style="list-style-type: none"> <li>a. El sistema no guarda els canvis</li> </ol> </li> </ol>
Fluxos alternatius	7. En cas d'error, el sistema ens informa amb un missatge
Postcondició	L'Usuari visualitza/edita el seu perfil

Cas d'ús	Visualitzar esdeveniment
Descripció	L'usuari pot visualitzar la descripció d'un esdeveniment i
Actor	Usuari autènticat
Precondició	L'usuari ha d'estar autènticat S'ha d'haver premut la miniatura de l'esdeveniment
Flux principal	<ol style="list-style-type: none"> <li>1. El sistema mostra la pàgina de l'esdeveniment</li> </ol>

Fluxos alternatius	2. En cas d'error, el sistema ens informa amb un missatge
Postcondició	L'Usuari visualitza l'esdeveniment

Cas d'ús	Xatejar amb un usuari
Descripció	L'usuari pot xatejar amb un usuari
Actor	Usuari autenticat
Precondició	L'usuari ha d'estar autenticat i tenir com a mínim 1 amic
Flux principal	<ol style="list-style-type: none"> <li>1.Usuari desplega el menú de xat de mà dreta</li> <li>2. L'usuari tria un usuari de la llista</li> <li>3. El sistema mostra (si existeix) els missatges anteriors amb l'usuari b</li> <li>4. L'usuari escriu un missatge a l'usuari b</li> <li>5. L'usuari b rep el missatge <ol style="list-style-type: none"> <li>5.1 Si l'usuari està offline, rebrà una notificació al pròxim login</li> </ol> </li> </ol>
Fluxos alternatius	6. En cas d'error, el sistema ens informa amb un missatge i/o no s'enviarà el missatge
Postcondició	L'Usuari ha xatejat amb un usuari b

Cas d'ús	Afegir comentaris en un esdeveniment
Descripció	L'usuari pot afegir comentaris en el perfil d'un esdeveniment
Actor	Usuari autenticat
Precondició	L'usuari ha d'estar autenticat i estar en el perfil d'un esdeveniment a

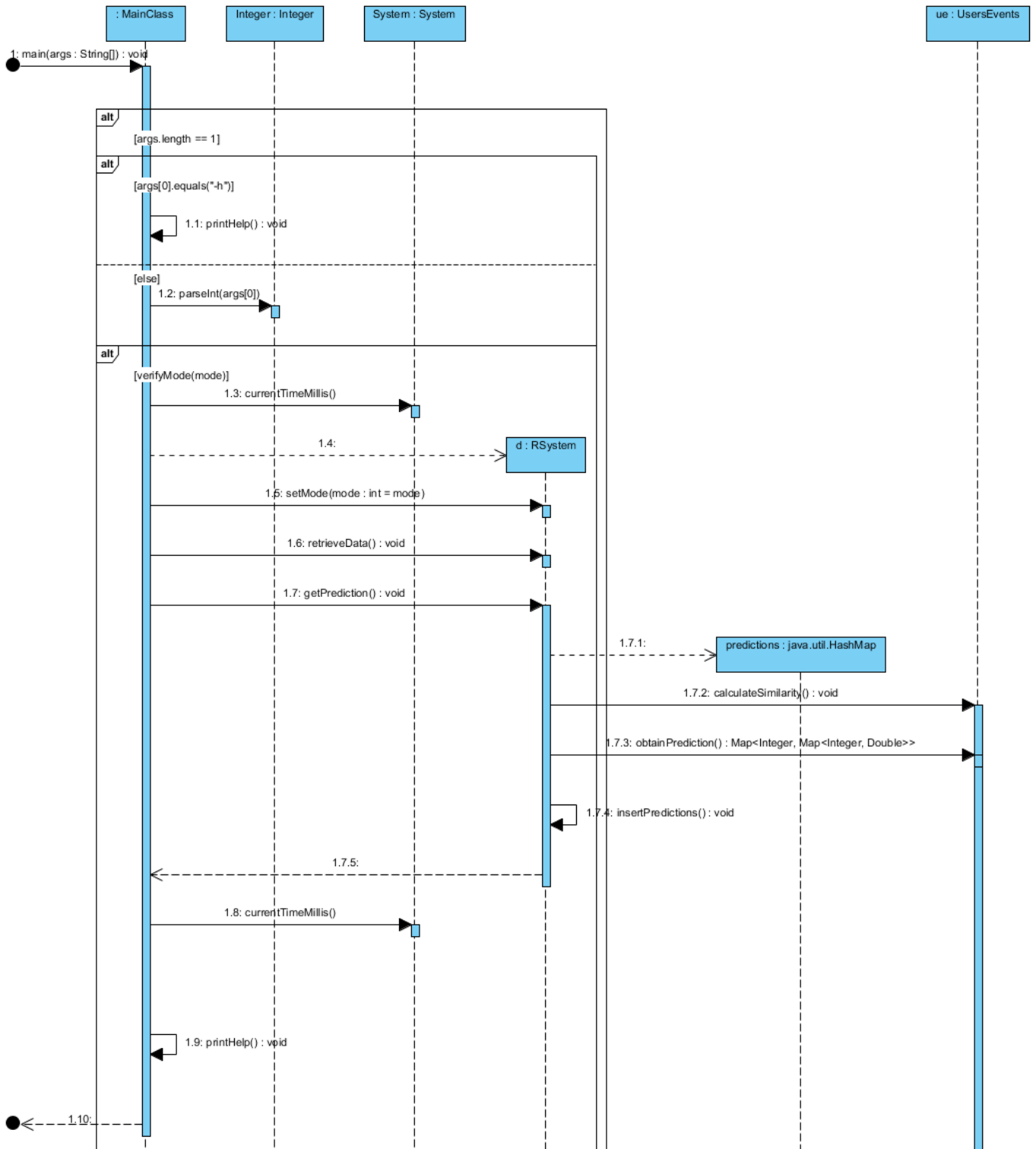
Flux principal	<ol style="list-style-type: none"> <li>1. L'Usuari pot llegir la llista de comentaris en el perfil de l'esdeveniment a</li> <li>2. L'usuari prem el botó "Afegir comentari"</li> <li>3. S'obre un popup on l'usuari pot escriure el comentari</li> <li>4. Si L'usuari tanca la finestra                         <ol style="list-style-type: none"> <li>4.1 No es publica el comentari</li> </ol> </li> <li>5. L'usuari escriu el comentari</li> <li>6 Si l'usuari prem "publicar"                         <ol style="list-style-type: none"> <li>6.1 El sistema publica el comentari</li> <li>6.2 El comentari surt publicat a la llista</li> </ol> </li> </ol>
Fluxos alternatius	<ol style="list-style-type: none"> <li>7. En cas d'error, el sistema ens informa amb un missatge i tanca el popup sense publicar el comentari</li> </ol>
Postcondició	L'Usuari publica un comentari a l'esdeveniment

Cas d'ús	Cercar Usuaris
Descripció	L'usuari pot editar el seu perfil
Actor	Usuari autèntificat
Precondició	L'usuari ha d'estar autèntificat

Flux principal	<ol style="list-style-type: none"> <li>1. L'usuari va a l'apartat "Usuaris" del menú dret</li> <li>2. Es despleguen dos subapartats</li> <li>3. Si L'usuari tria "Amistats" <ol style="list-style-type: none"> <li>3.1 El sistema mostra un llistat de tots els usuaris amb qui manté una relació d'amistat</li> <li>3.2 L'usuari pot escriure a la barra de cerca de la part superior</li> <li>3.3 Si l'usuari prem "cerca"</li> <li>3.4 El sistema mostra tots els usuaris-amics de la plataforma que coincideixen amb el text entrat en nom+cognoms</li> </ol> </li> <li>4. Si L'usuari tria "Tots els Usuaris" <ol style="list-style-type: none"> <li>4.1 El sistema mostra un llistat de tots els usuaris de la plataforma</li> <li>4.2 L'usuari pot escriure a la barra de cerca de la part superior</li> <li>4.3 Si l'usuari prem "cerca"</li> <li>4.4 El sistema mostra tots els usuaris de la plataforma que coincideixen amb el text entrat en nom+cognoms</li> </ol> </li> </ol>
Fluxos alternatius	<ol style="list-style-type: none"> <li>5. En cas d'error, el sistema ens informa amb un missatge i no mostra cap usuari</li> </ol>
Postcondició	L'Usuari visualitza/cerca usuaris a la plataforma

### 8.3 Diagrames de seqüència

A continuació podem veure el diagrama de seqüència del sistema recomanador d'esdeveniments.

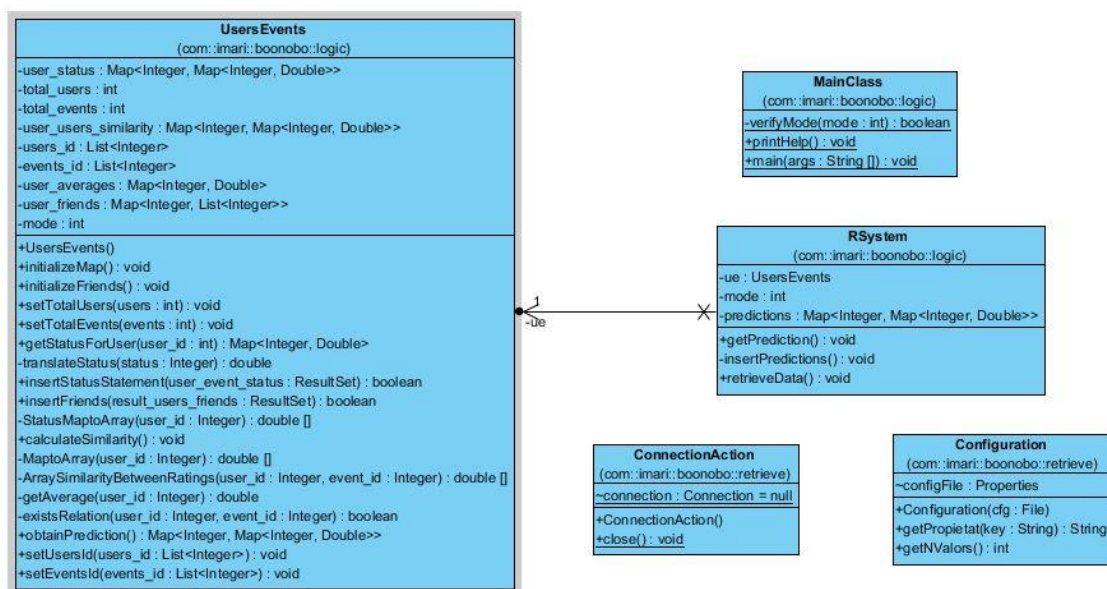


II-lustració 11 - Diagrama de seqüència de profunditat 2 del Sistema Recomanador

## 8.4 - Diagrames de classe

En aquest apartat, s'explicaran els diagrames de classes que es va seguir per realitzar la implementació tant del sistema recomanador com de l'aplicació.

### 8.4.1 - Diagrama del Sistema Recomanador



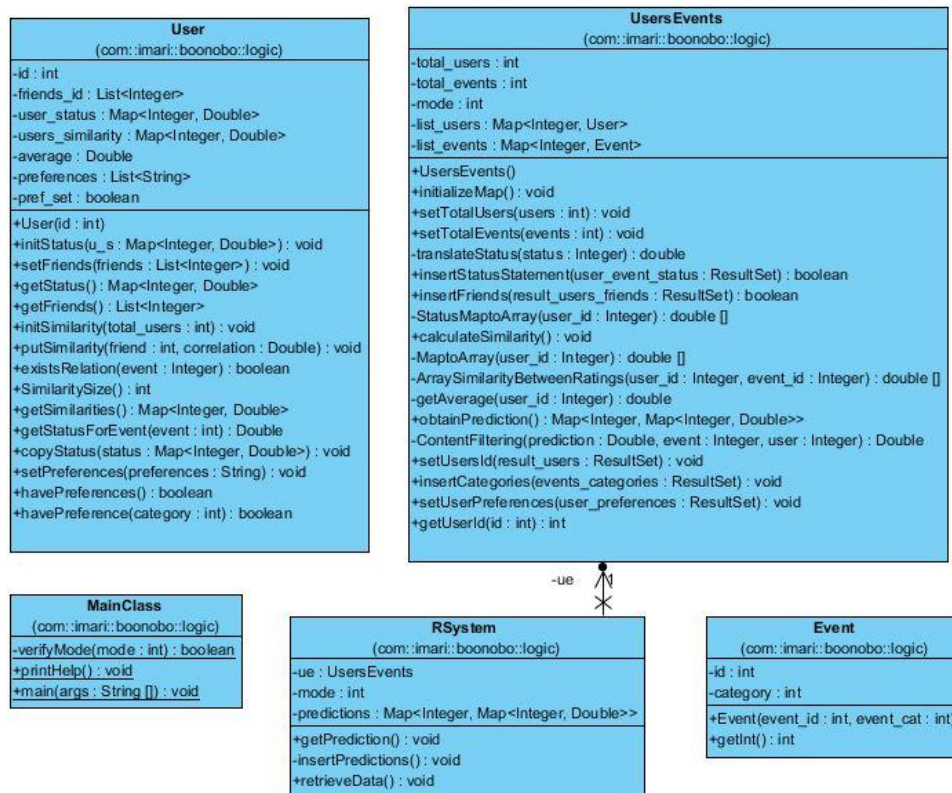
### II·lustració 12 - Diagrama versió 1 del sistema recomanador

El sistema recomanador el podem dividir en dos paquets o grups segons responsabilitats: tenim la lògica del sistema i el procés de connectar amb la base de dades per recol·lectar o/i afegir les dades. A la lògica hem volgut distingir 3 nivells de responsabilitat: tenim el programa principal que rep l'entrada del procés i comprova que estigui bé i afegeix un Sistema Recomanador (RSystem) que realitzarà tot el procés de manera que, en forma de caixa negra, el programa principal no sap que realitza aquesta classe.

RSystem coneix en tot moment quines classes resolen els passos que ha de seguir per desenvolupar la seva tasca en plenitud. De manera que primer crida al sistema “retrive” (ConnectionAction i Configuration) que funciona de pont entre les dades i el sistema recomanador. Un cop ha obtingut les dades, crea un objecte UsersEvents on la seva principal tasca serà agafar aquestes dades i processar-les (de manera que aquest nivell

intermedi que representa el RSystem no sap realment quines operacions es fan sobre les dades.

La classe UsersEvents representa tot el conjunt de dades i operacions que tenim dels Usuaris i dels Esdeveniments tant pròpies com compartides; un cop acabat el seu desenvolupament **s'ha aplicat una refactorització Extract Class** doble extraient dues classes noves: User i Event.



Aquest refactor té com a motiu principal diferenciar els dos ens de treball amb els seus respectius atributs, i resoldre les operacions pròpies sobre les dades; podríem dir doncs que és una refactorització per responsabilitat i per modularitat. És curiós també que aquest refactor no només ens ha millorat l'entesa del codi, la modularitat i la representació de responsabilitats: també ens ha millorat el rendiment. Per què?

El motiu és que en la versió anterior treballavem amb estructures de dades feixugues com són Maps guardats com a valors d'altres Maps. Això ens baixa el rendiment, tot i que són les estructures de dades més ràpides en obtenció, pel temps de lectura i escriptura que perdíem. S'ha millorat en el moment que aquestes dades es representen com a objectes i la referència de les dades, entenenent com a punters de memòria, és millor si en comptes de moure i borrar dades, modifiquem els punters d'aquestes. És ara User qui realitza els càlculs propis d'un sol usuari i s'ha deixat a UsersEvents només la responsabilitat de fer

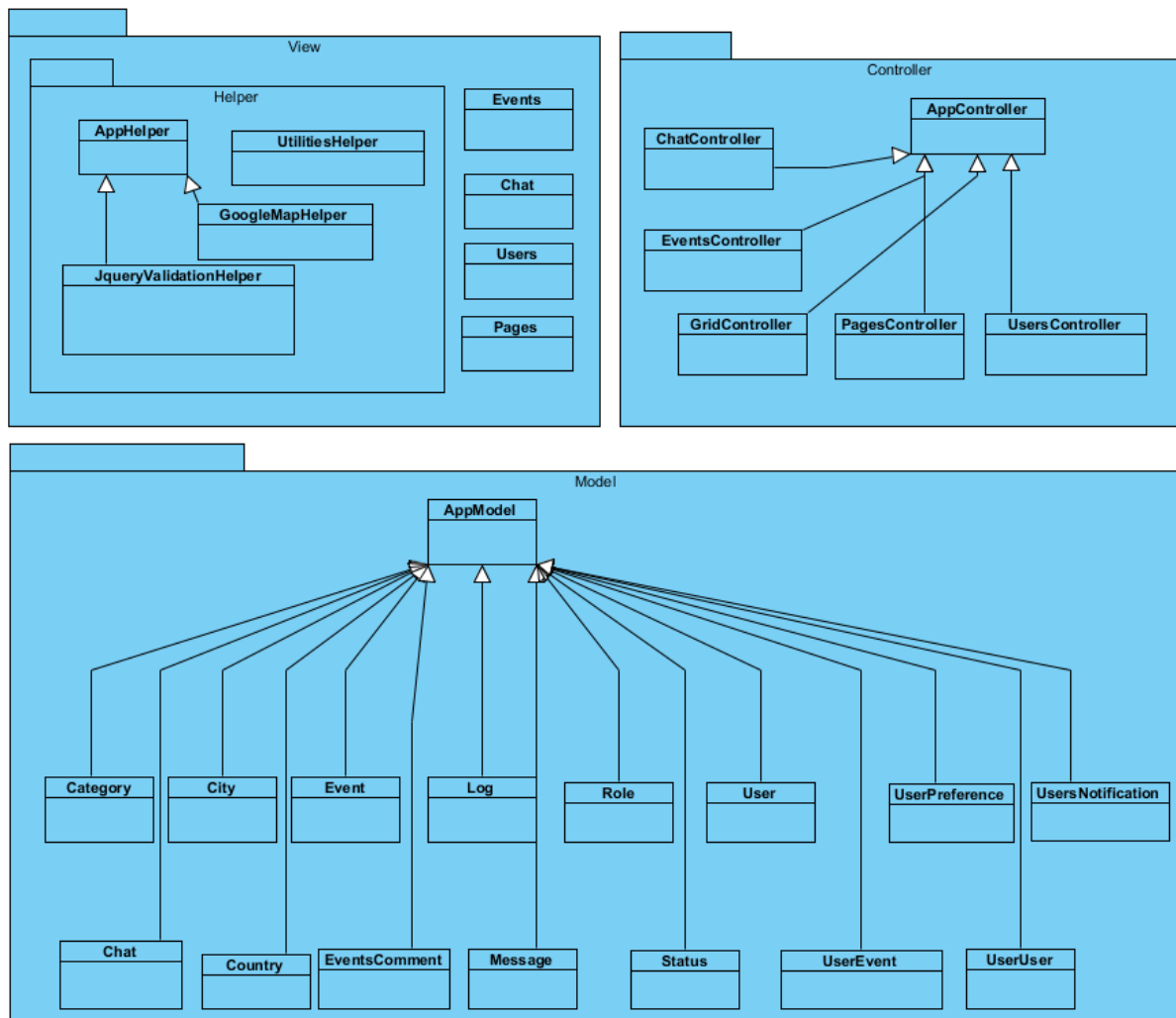
càlculs amb tots el conjunt de dades en sí.

A les hores la pregunta és, i perquè no realitza aquestes operacions el RSystem? Si bé aquest és qui acaba executant aquestes operacions, la decisió de divisió de responsabilitats no concordaria amb aquesta possibilitat. RSystem executa l'abstracció de dades, però finalment no sap com les treballa la classe UsersEvents, pel que no podria aplicar (ni hauria, com hem comentat) operacions sobre aquestes.

Finalment UsersEvents retorna un seguit d'arrays de dades, que s'assemblen a les obtingudes d'entrada, a RSystem que és l'encarregat de cridar el procés que les inserirà a la base de dades.

Com hem comentat, s'ha utilitzat la divisió de tasques i classes per responsabilitat, però encara quedaria aplicar un patró més (es reconeix ara durant la redacció d'aquesta memòria) que ajudaria a tenir les dades més ben encapsulades.

#### 8.4.2 - Diagrama de l'aplicació web Boonobo



Il·lustració 13 - Diagrama de l'aplicació web

En aquest diagrama podem apreciar clarament el patró MVC que implementa CakePHP. S'ha decidit prescindir de moltes classes de llibreries i *Helpers* que proporciona el Cake per basar-nos únicament en les que representen la web en sí.

Per començar, cal apreciar (com veurem més endavant al diagrama de la base de dades) que existeixen tants models com taules a la base de dades. Les taules tenen la peculiaritat d'acabar els seus noms en "s" i Cake transforma els models en la paraula singular. S'inclouen també les taules de relació.

Després tenim les capes de controlador i vista, aquestes capes han d'estar estrictament relacionades, ja que a cada controlador li correspondrà un conjunt de classes anomenades com ell. El sistema intern de Cake ja fa la transformació de noms pel que ja sap que *EventsControllers* equival a les vistes incloses dins de *Events*. Cada mètode del controlador que vulguem visualitzar o que utilitzem per una vista en concret, haurà de tenir el seu homòleg a la corresponent carpeta de vista amb el *nomdelmètode.ctp* (no s'inclouen en el diagrama de classes per ser més fixers que no pas classes en sí).

Al diagrama apareixen varis *Helpers* utilitzats fortament a la Vista: *JqueryValidation*, *Google Maps* i *Utilities* són un seguit de classes afegides al *cake* (disponibles a la seva *web* de documentació) per millorar el tractament de dades.

Finalment com podem observar, totes les classes de Model estenen (és a dir, hereten) d'una classe principal *AppModel* i totes les classes del Controlador, d'*AppController*. És en aquestes classes on es definiran mètodes que heretaran tots.

## 8.5 - Diagrama d'Entitat-Relació (Base de dades)

Cal tenir en compte des de bon inici, que la base de dades és la mateixa tant pel sistema recomanador com per l'aplicatiu web. Es nodreixen ambdós de la mateixa base de dades i per tant, hi haurà taules que hauran estat modificades per servir a un o a l'altre.

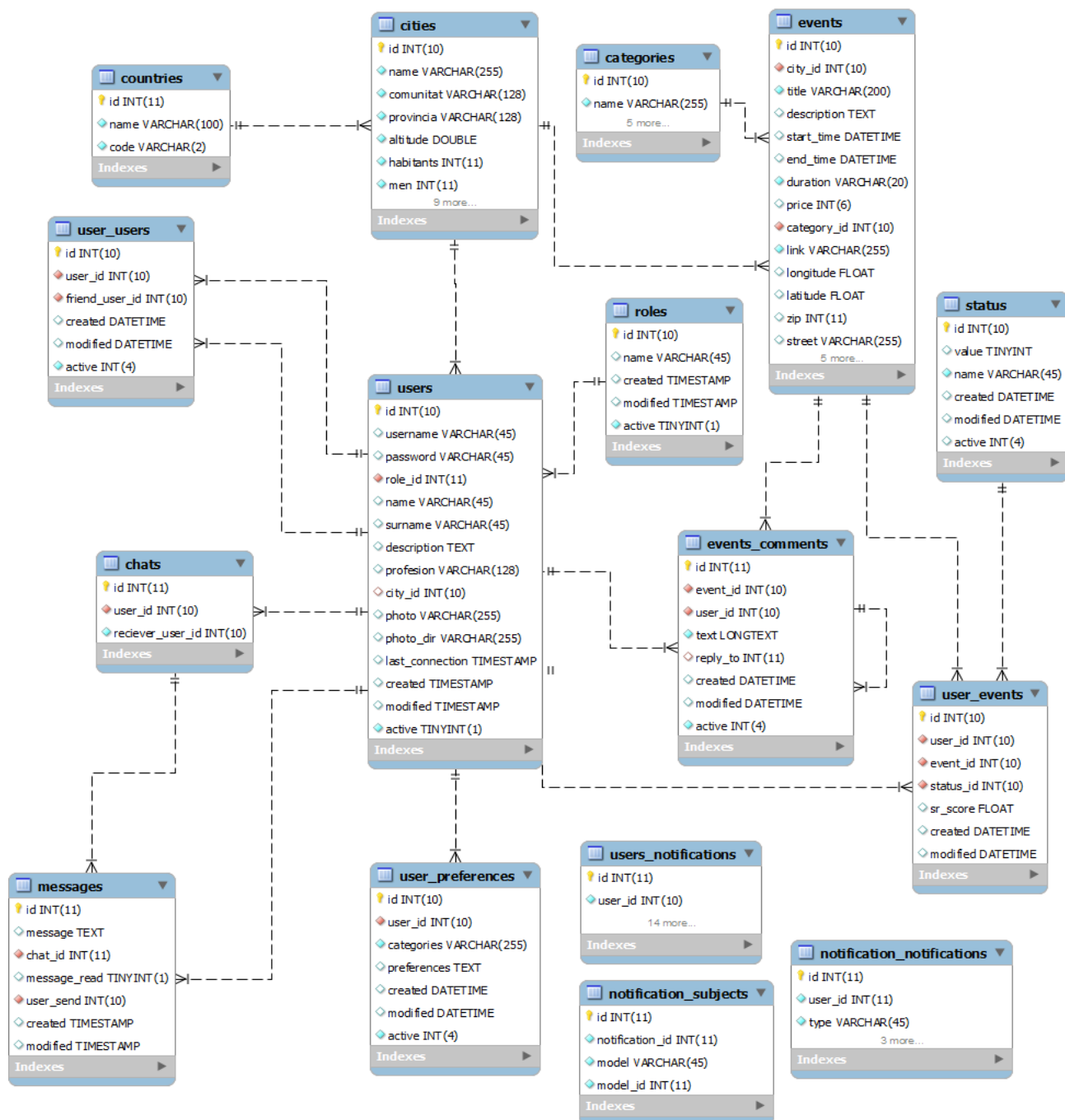
També cal tenir en compte que l'aplicatiu web parteix d'un projecte anterior que és el Boonobo original, per tant les taules de les bases ja ens venen donades com a dades prèvies al projecte.

Per tant, en aquest apartat explicarem els canvis respecte el model ER que ja se'ns donava des de bon inici.

- Totes les claus primàries compostes d'alguna relació (exemple *user\_events*) s'han hagut de substituir per una clau primària única generada a partir d'Auto increment (CakePHP no accepta claus compostes com a claus primàries).
- S'ha afegit la columna "*active*" i "*modified*" a les taules necessàries pel correcte

funcionalment de quèries del CakePHP

- S'ha creat la taula *user\_preferences* per a identificar un conjunt de preferències per a l'usuari. Conté una clau primària *id*, clau forana *user\_id* i un seguit de columnes de les quals finalment només hem utilitzat *categories* (que representa un array de categories). Les preferències (que més endavant es convertiran en paraules clau) s'han deixat a la taula per a un treball futur sobre el Sistema Recomanador
- Les taules *user\_notifications*, *notification\_notifications* i *notification\_subjects* han estat afegides pel correcte funcionament de l'aplicatiu web, no obstant, no estaran presents en el treball futur per ser requeriments del CakePHP
- Les taules *chats* i *messages* que relacionen als usuaris i les seues missatges en xat es creen per desenvolupar una funcionalitat del *template* de *bootstrap* que utilitzem i també estaran presents en el futur treball amb les mateixes característiques (en principi)
- Es crea la taula *status* per identificar tots els possibles estats que pot tenir un usuari amb un esdeveniment.
- Es crea la taula *user\_events*, clau en aquest projecte, que relaciona els usuaris amb els esdeveniments i se'ls pondera amb un estat. Aquesta relació serà única, encara que no puguem utilitzar clau primàries compostes, i serà modificada a cada actualització d'estat.



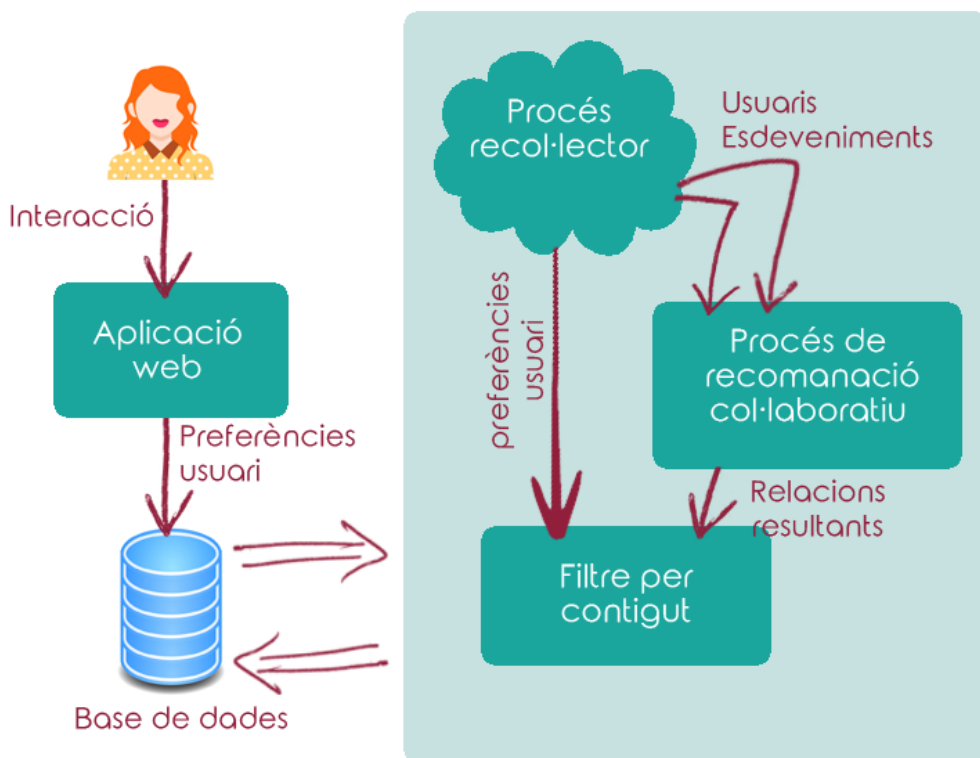
## II·lustració 14 - Diagrama ER de la Base de Dades

## 9. Implementació i proves

En aquest capítol es detallarà l'estructura de l'algorisme del sistema recomanador implementat en aquest projecte, així com la divisió dels apartats de l'aplicatiu web i les proves que s'han realitzat sobre el projecte.

### 9.1- Estructura del sistema recomanador

Com s'ha explicat en apartats anteriors, el sistema recomanador d'aquest projecte es basa en un híbrid on la base és un sistema col·laboratiu que a posteriori s'aplica un filtre per contingut amb el resultat d'aquest. A continuació un esquema del seu funcionament:



**Il·lustració 15 - Esquema del funcionament del Sistema Recomanador**

L'aplicatiu web recull totes les dades necessàries de l'usuari per poder-les treballar al complet.

L'usuari, des del seu perfil, pot concretar quines preferències de categories té, per defecte es ponderaran a totes per igual. També podria triar la ciutat però, a l'haver-hi només esdeveniments de Girona en el *dataset* original, no es deixa canviar. De l'usuari també recol·lectem la seva activitat de manera que puntuem cada relació d'un usuari amb un esdeveniment segons aquests criteris:

Descripció	Valor
Llegit	1
Assistirà	3
M'agradaria	2
No aniré	-1
No vull veure'l més	-2
Recomanat	5

Cal tenir en compte que quan les dades són recol·lectades per enviar-les al sistema recomanador, l'últim valor com és "recomanat" no es té en compte, per tant totes les relacions ponderades d'aquesta manera seran menyspreades pel sistema recomanador. Per què? Doncs perquè són la sortida d'un procés anterior.

Quan un usuari llegeix un esdeveniment, es crea el lligam entre usuari-esdeveniment. Si ja existia aquest lligam, no es fa res ja que es considera que la resta tenen més "poder" encara que tinguin valor més petit.

L'usuari pot triar dins de l'esdeveniment "Assistiré", "No assistiré" i "M'agradaria". Per ordre de rellevància, "No assistiré" se'l classifica com a negatiu, distingint-lo així de la resta d'estats que tenen un aire "positiu".

Un cop el sistema recomanador aconsella aquest esdeveniment a l'usuari, es crea una relació amb valor 6. Si l'usuari el veu a la llista de recomanats i no li agrada aquesta recomanació, la primera opció que veurà, abans i tot de llegir-lo, serà "No vull veure'l més".

Si l'usuari no interacciona en cap moment amb l'esdeveniment, de manera directa o indirecta, no existirà la relació entre ells.

Les preferències de l'usuari seran dades explícites, així com la tria dels estats; l'opció de llegir-lo o no serà implícit (del comportament de l'usuari).

Un procés amb *background* executa el procés del sistema recomanador i aquest recol·lecta

4 taules essencials de la base de dades: usuaris, esdeveniments, usuari-esdeveniment i usuari-usuari (que representa l'amistat i segons en quin model executem el SR, la utilitzarem per acotar el veïnat).

Es calcula doncs la similitud entre els usuaris a través de la correlació de Pearson. **És possible que aquesta correlació doni NaN en Java, per què?** El motiu és quan a la Correlació dividim per 0, i és que no es troben coincidències d'esdeveniments puntuats pels dos; és bo substituït aquest NaN per 0? Molts manuals recomanen que no, ja que aquest NaN tant pot venir d'un 0 com d'una arrel quadrada de negatiu. També pot portar problemes de representació de la similitud (si resulta que 0 forma part de la nostra escala de ràtings, pot portar a una confusió). En aquest cas s'ha optat per triar la via de substituir-ho per 0.0 tenint en compte els debats i desacords que hi ha en aquesta decisió.

Un cop tenim les similituds, es calcula la predicció de cada usuari per cada esdeveniment però amb compte: si un usuari te menys de 5 relacions amb esdeveniments, automàticament **el sistema recomanador només li recomanarà el top 20 d'esdeveniments**. Per què? El problema està en la fiabilitat de les recomanacions quan un usuari ha interaccionat tant poc, i són 5 segons com marca el professor Joseph Konstant perquè es creu que és la frontera d'aquest esdeveniment.

Si l'usuari ja té una relació, que no sigui recomanat, s'obviarà aquest coeficient de recomanació, altrament s'actualitzaria.

Un cop tenim tot el conjunt de puntuacions, cal ponderar-les segons les dades explícites que ens ha donat l'usuari: les seves preferències. És el moment doncs del filtratge per contingut.

**Quina ponderació cal donar?** Els experts defensen que cada sistema és diferent (pot ser que en comptes d'obtenir el possible valor de l'usuari amb l'esdeveniment, els ràtings fossin binaris, o de 0 a 5) i per tant només amb assaig-error es pot obtenir el número perfecte a afegir o restar (en el nostre cas). Però, mai superant la diferència numèrica entre esgraons de ràting (per tant mai superant 1 punt).

Finalment s'ha decidit que si l'usuari te introduïdes les seves preferències, i aquestes concorden amb l'esdeveniment en concret, se sumará 0.5 a la puntuació; altrament es repta 0.5. Si no té les preferències afegides, no es farà res.

Tenint en compte això, es mostraran com a recomanats i s'inseriran a la base de dades, tots els esdeveniments que tinguin una puntuació  $\geq 1.8$  per tant que es trobin entre "Llegit" i "m'agradaria" més a prop del segon.

## 9.2 - Els apartats de l'aplicatiu web

L'aplicatiu web té diferents apartats des d'on recollim tota la informació. La informació explícita es recull al perfil de l'usuari (Perfil->Editar Perfil).

La informació implícita es recull a través de la interacció de l'usuari amb els esdeveniments.

Per fer-ho pot entrar a varis apartats:

- Inici/Esdeveniments->Recomanats: Aquí l'usuari pot veure tots els esdeveniments que se li recomanen a l'usuari; en cas que el procés del SR encara no estigui engegat li mostrarem els 5 esdeveniments amb més usuaris assistents.
- Esdeveniments->Top 20: El sistema fa un recompte de tots els assistents per cada esdeveniment i ens mostra un conjunt dels 20 amb més assistents.
- Esdeveniments->Tots els esdeveniments: El sistema mostra tots els esdeveniments que tenim disponible a tot l'aplicatiu
- Esdeveniments-> Random: El sistema genera un seguit de ids aleatòries per obtenir de la base de dades: un total de 20

En tot moment els esdeveniments són de totes les categories sense filtrar i tenint en compte que el data d'inici  $\leq$  moment consulta i en cas contrari, que la data de finalització  $\geq$  moment de la consulta.

## 9.3 - Fases de proves del funcionament

La característica principal d'aquest projecte és que ja ha estat testejat amb usuaris reals. Com s'ha comentat al inici, la metodologia àgil permet anar afegint noves funcionalitats i corregir ja les implementades gràcies al feedback dels usuaris que la proven. Així com es pot apreciar als *sprints*, durant cada setmana s'han incorporat bugs i millores al disseny. Els dividim en dos grups:

- **Els errors de la pàgina web:** són errors típics al traslladar una aplicació local a un servidor remot, amb el *user-testing* es detecten errors ocasionats per la navegació típica d'un usuari aliè al desenvolupament i d'una manera totalment lògica per a ell. Es resolen problemes de navegació i càrrega d'imatges i contingut que encara no es troba en la *cache* del navegador de l'usuari. Es recullen també recomanacions de funcionalitat que seran traslladades al projecte principal Boonobo fora d'aquest PFG.
- **Els errors del sistema recomanador:** No sempre són perfectes i hi ha hagut grans canvis entre la primera versió (només el sistema col·laboratiu) i la segona versió (s'inclou el filtre per contingut). Després s'ha aprofitat el *feedback* dels usuaris per

calibrar millor la puntuació que obtenim després de sumar/restar la ponderació per contingut. També ha ajudat el *feedback* a l'hora de troba relacions sobre el perquè d'esdeveniments recomanats que finalment no es corresponen amb els gustos de l'usuari.

Aquests errors han estat recopilats després d'enviar un manual als usuaris pel que fa a quins passos havien de seguir i proporcionar-los una direcció web on enviar els errors i suggeriments.

## 10. Implantació i resultats

### 10.1 - Implantació

L'aplicació està de moment disponible al servidor web d'Amazon sota el domini <http://boonobo.es> però un cop finalitzat el projecte pare del que parteix aquest PFG, el contingut serà substituït per l'aplicatiu final. No requereix cap tipus d'instal·lació per ser utilitzada. El sistema recomanador és ocult per a l'usuari, treballa amb *background* al servidor Amazon i mentre la base de dades no canviï, continuarà funcionant igual.

### 10.2 - Resultats

Encara que l'aplicatiu només sigui la interfície per on recollim les dades, també forma part en sí d'aquest projecte i juntament amb com es mostra la sortida del sistema recomanador, conformen els resultats que a continuació es mostren classificats segons els requeriments funcionals.

#### 10.2.1- Login

Quan obrim l'aplicació se'ns redirigeix a aquesta pantalla per iniciar sessió [Figura 16]. Si afegim un *email* mal format o una contrasenya menor a la necessària per a registrar-se, dóna error abans d'enviar la informació [Figura 17-18]. Si les dades d'autenticació són incorrectes ensenya un altre missatge d'error [Figura 19] , altrament s'autentifica correctament i ens redirigeix a la pàgina inicial



## Accés al compte

### II-lustració 16 - Formulari de login

\* Entra una direcció e-mail vàlida

\* Ha de tenir com a mínim 5 caràcters

### II-lustració 17 - Missatges en cas d'error

Usuari o contrasenya incorrecta, si us plau,  
torna-ho a provar

### II-lustració 18 - Missatge en cas d'usuari incorrecte



**Il·lustració 19 - Pàgina principal en primer inici**

### 10.2.2- Registrar-se

Premem el botó “registrar-se” i la pantalla canvia per mostrar-se el formulari de registre [Figura 20]. Omplim el formulari que verifica, i sinó dóna error, que el mail estigui ben format, que la contrasenya tingui com a mínim 5 caràcters i que les dues contrasenyes coincideixin [Figura 21-22]. Un cop s’accepta el botó d’ “Accepto” i tot es correcte, s’habilita el botó de “Registre” per enviar el formulari. Si l’usuari es registra correctament s’informa amb un missatge verd al l’usuari [Figura 23] altrament en vermell.

## Registra't És gratis i sempre ho serà

Nom	Cognoms
Email	
Contrasenya	Confirma contrasenya

☐ **Accepto**

Quan cliquis **Registre** estàs acceptant els nostres [Termes i Condicions](#) inclosos en aquesta web, inclòs l'ús de Cookies.

**Registre**

**Enrere**

### II·lustració 20 - Formulari de registre

mailincorrecte	
Email no vàlid	
...	..

Les contrasenyes no coincideixen!

### II·lustració 21 - Camps de registre en cas d'error

Registra't És gratis i sempre ho  
serà

Registre	Deprova
mail@correcte.es	
Email correcte	
.....	.....

Les contrasenyes coincideixen!

☒ **Accepto**

Quan cliquis **Registre** estàs acceptant els nostres  
Termes i Condicions inclosos en aquesta web, inclòs  
l'ús de Cookies.

**Registre** **Enrere**

II-lustració 22 - El formulari pre-valida les nostres dades



**Usuari registrat correctament**

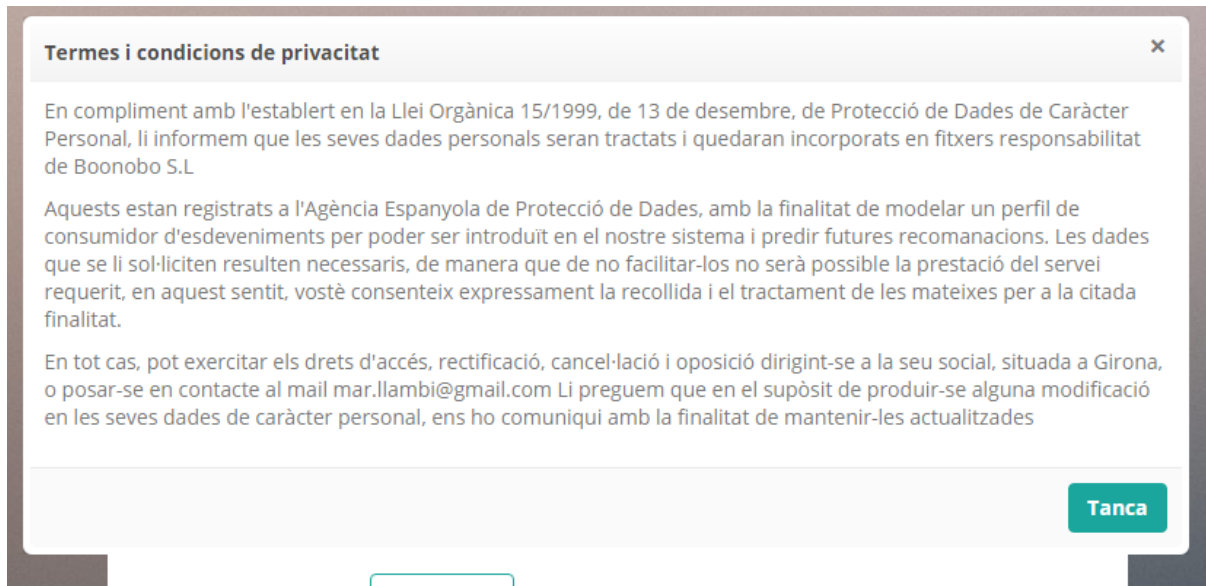
Correu electrònic
Clau

**Entrar** **Registra't**

II-lustració 23 - L'usuari s'ha registrat correctament

### 10.2.3- Llegir termes i condicions

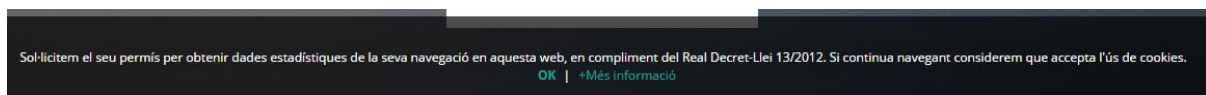
Durant el registre, es poden llegir els termes i condicions, si premem sobre el terme “Termes i condicions” del formulari de registre i llavors es desplega un *popup* com el que es mostra [Figura 24]. Un cop llegides podem tancar el *popup* per continuar amb el registre.



II-lustració 24 - Termes i condicions de l'aplicació

### 10.2.4- Llegir/acceptar política de cookies

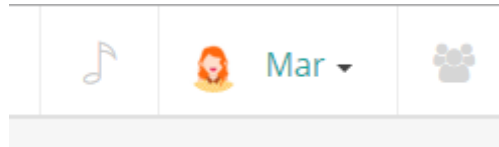
Durant tota l'experiència de navegació de l'usuari a la web, se l'informa de l'ús de *cookies* tal com marca la legislació vigent. Es proporciona un botó de “OK” per fer desaparèixer aquest missatge o un de “+ Informació” que redirigeix a la web per defecte de l'Estat sobre la llei (ja que no s'utilitzen *cookies* pròpies que descriure).



II-lustració 25 - Política de cookies present en totes les pàgines de l'aplicació

### 10.2.5- Tancar sessió

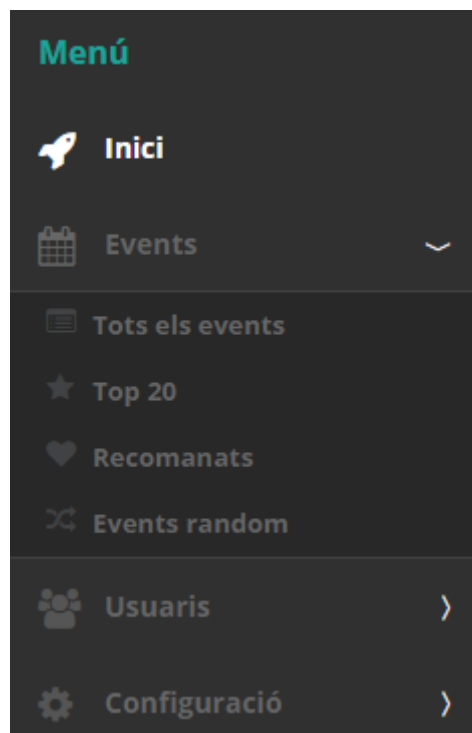
Al menú superior, junt amb el nom de l'usuari [Figura 26], si acostem el ratolí es desplegarà dues opcions “Perfil” o “Sortir” si premem la segona, es tanca la sessió i se'ns redirigeix a la pàgina de Login.



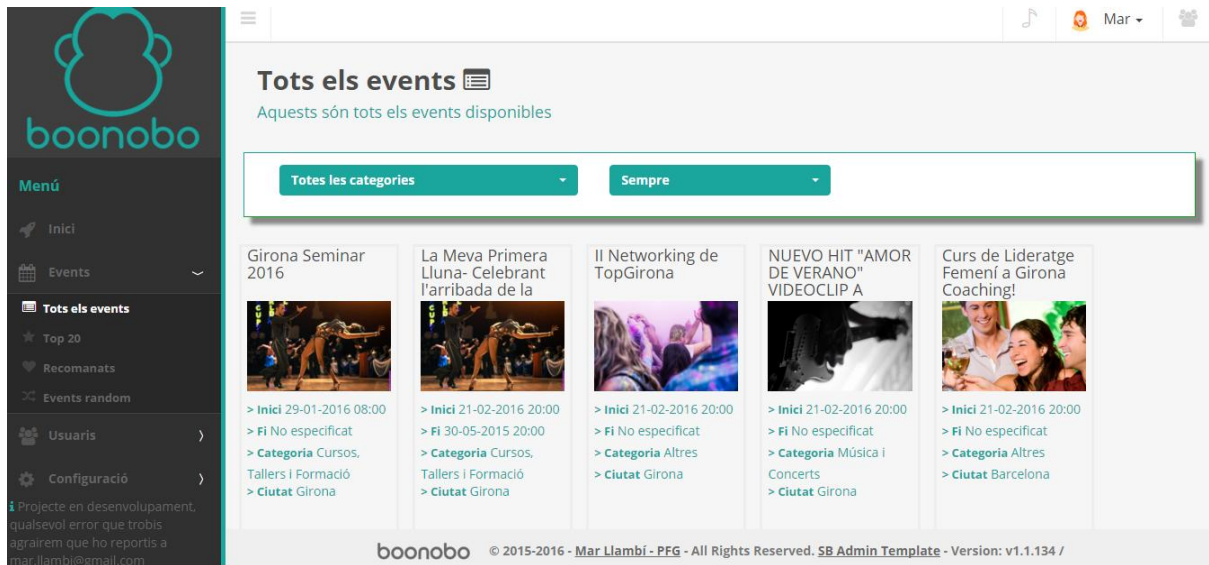
II-lustració 26 - Menú superior dret

### 10.2.6- Visualitzar esdeveniments

Al menú de mà esquerre podem veure un apartat com és “Events” i cliquem, s'obre un desplegable amb diverses opcions [Figura 27]. En aquest cas triem l'opció “Tots els events”. L'aplicatiu ens redirigeix a una pàgina on es llisten tots els esdeveniments per qualsevol categoria en qualsevol moment del temps [Figura 28] A partir d'aquí podem filtrar per categoria o per temps com mostren els filtres [Figura 29-30].



II-lustració 27 - Menú esquerre amb l'apartat Events desplegat



II-lustració 28 - Apartat Tots els esdeveniments



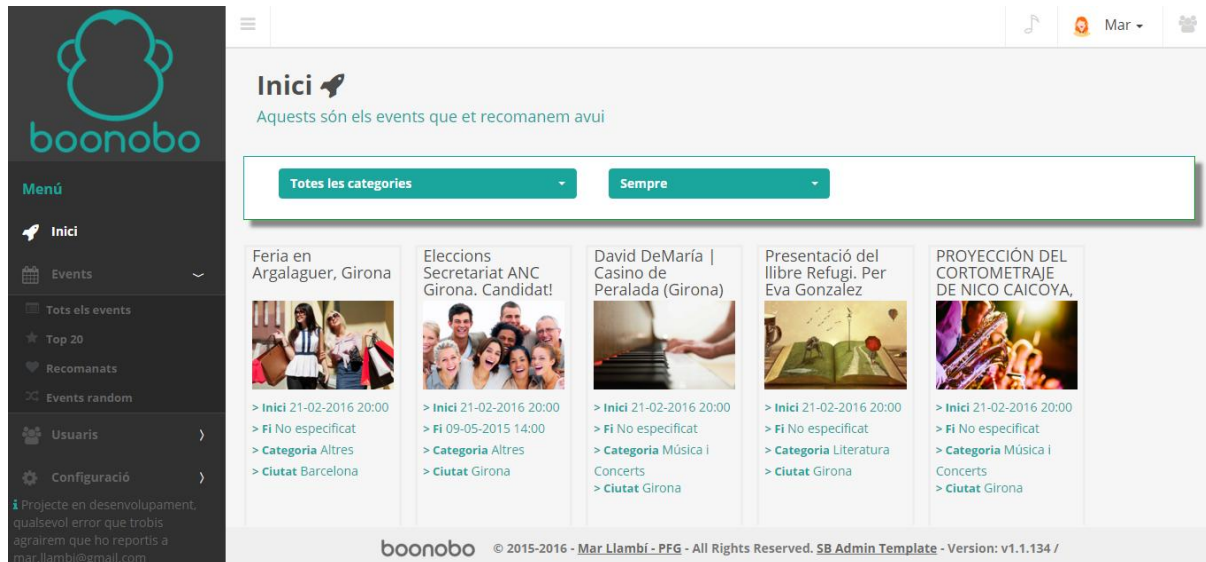
II-lustració 29 - Filtres disponibles per esdeveniments



II-lustració 30 - Mostra del filtre per categories de l'aplicació

### 10.2.7- Visualitzar esdeveniments recomanats

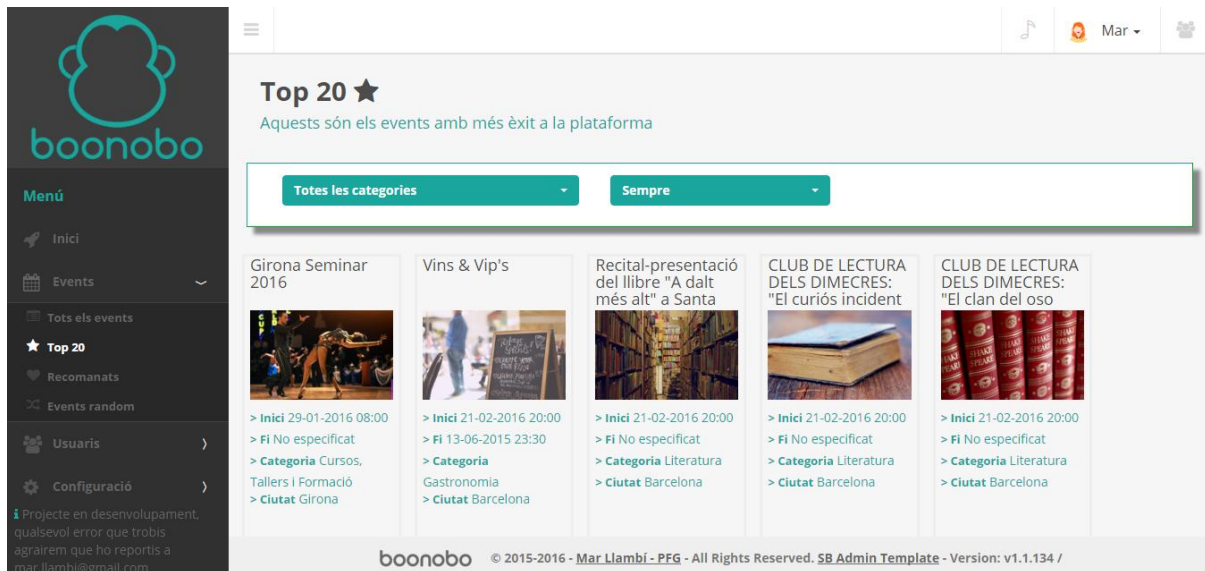
En aquest cas anem al menú esquerre i, al desplegable d'esdeveniments, marquem l'opció "Esdeveniments recomanats". També podem accedir-hi anant a l'"inici", ambdós redirigeixen a la mateixa pàgina [Figura 31]. Novament podrem filtrar per categoria i franja temporal.



**Il·lustració 31 - Apartat Inici (correspon també a recomanats)**

### 10.2.8- Visualitzar Top 20 d'esdeveniments

En aquest cas ens situem al menú esquerre altre cop i al menú, a l'apartat d'esdeveniments, podem triar el sub-apartat "Top 20". Un cop aquí l'aplicatiu ens redirigeix a una pàgina amb els esdeveniments amb més assistents [Figura 32]. Altre cop es pot filtrar per categoria i franja de temps.



II-lustració 32 - Pàgina de Top 20 de l'aplicació

### 10.2.9- Afegir amiatat

Un cop visualitzem el perfil d'un usuari, si no és amic nostre se'ns indica la opció d'"Afegir amiatat" a la banda dreta del perfil de l'usuari [Figura 33] i se'ns informa que al no ser amics, no podem veure tot el perfil. Premem el botó per afegir-lo com amiatat i un missatge en verd ens avisarà que s'ha afegit correctament [Figura 34] altrament si hi hagués algun error, el missatge seria vermell. Un cop afegit com amiatat, ja podrem veure el seu perfil complet [Figura 35]



II-lustració 33 - Perfil de l'usuari sense haver-hi relació d'amiatat



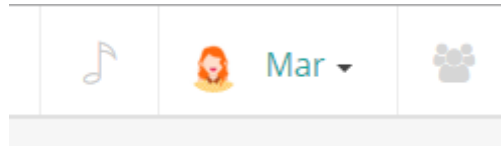
II-lustració 34 - Confirmació de la pàgina a l'afegir amistat



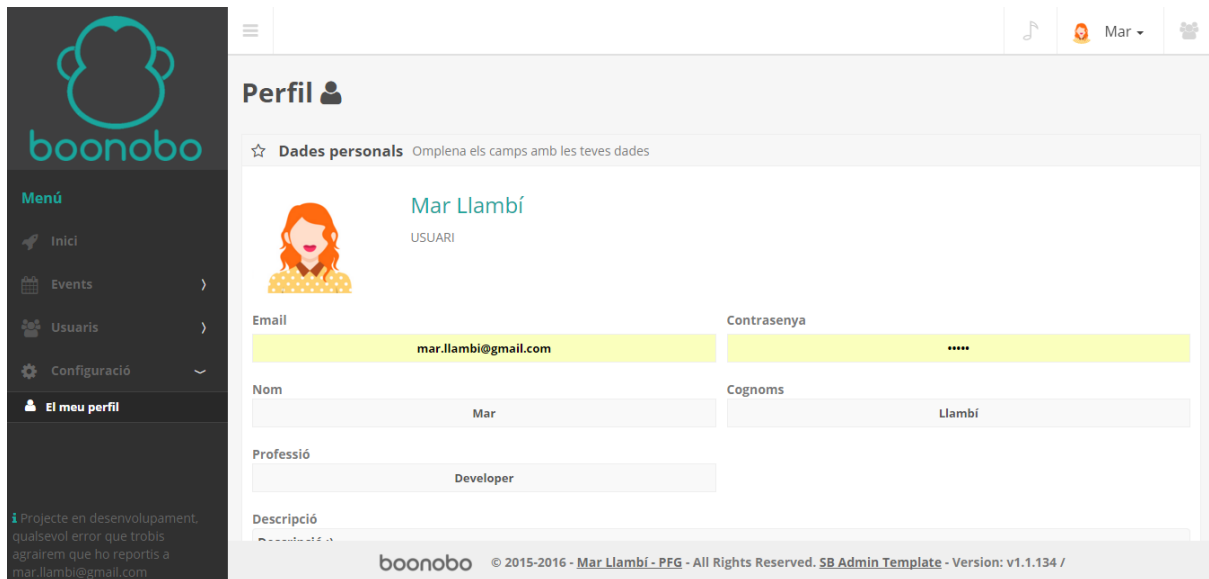
II-lustració 35 - Perfil de l'usuari al ser afegit com amistat

### 10.2.10- Editar perfil

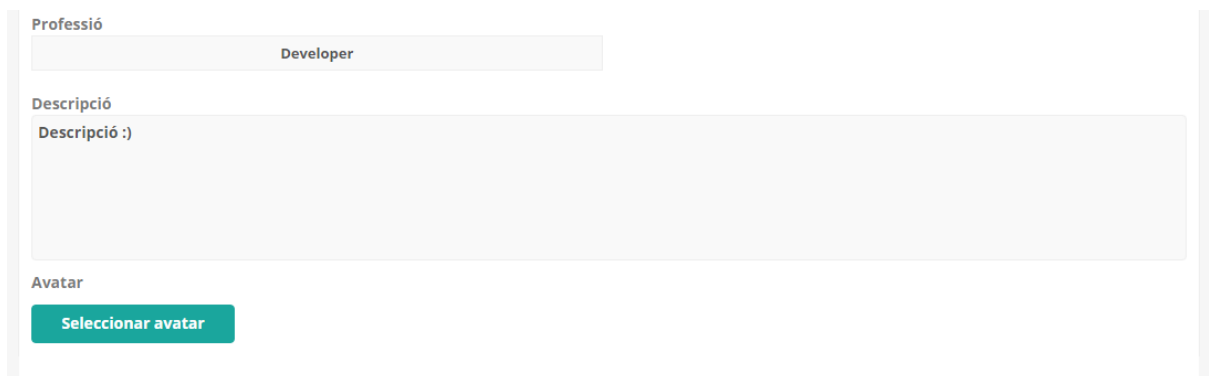
Al menú superior, on s'indica el nostre nom, recordem que s'obre un desplegable [Figura 36]. Allà podrem triar l'opció de "Perfil" i l'aplicatiu ens redirigirà al nostre perfil [Figura 37] on podem editar la informació bàsica (Nom, Cognoms, professió, informació..). Tanmateix, podem canviar el nostre avatar de perfil si premem el botó "Seleccionar avatar" [Figura 38]. Se'ns obrirà un *popup* [Figura 39] i podrem seleccionar la imatge que vulguem. Si continuem amb l'edició del perfil i seleccionar les categories que entren en les nostres preferències [Figura 40], també podem visualitzar la llista d'esdeveniments que hem assistit [Figura 41] i veure un mapa [Figura 42] amb la nostra localització (que recordem que per defecte és Girona). Per últim, per guardar els canvis cal anar a buscar i prémer el botó "Guardar canvis" [Figura 43]. Si tenim les dades guardades al navegador es guardaran els canvis [Figura 44], altrament ens obligarà a re-introduir la contrasenya a la part superior del perfil.



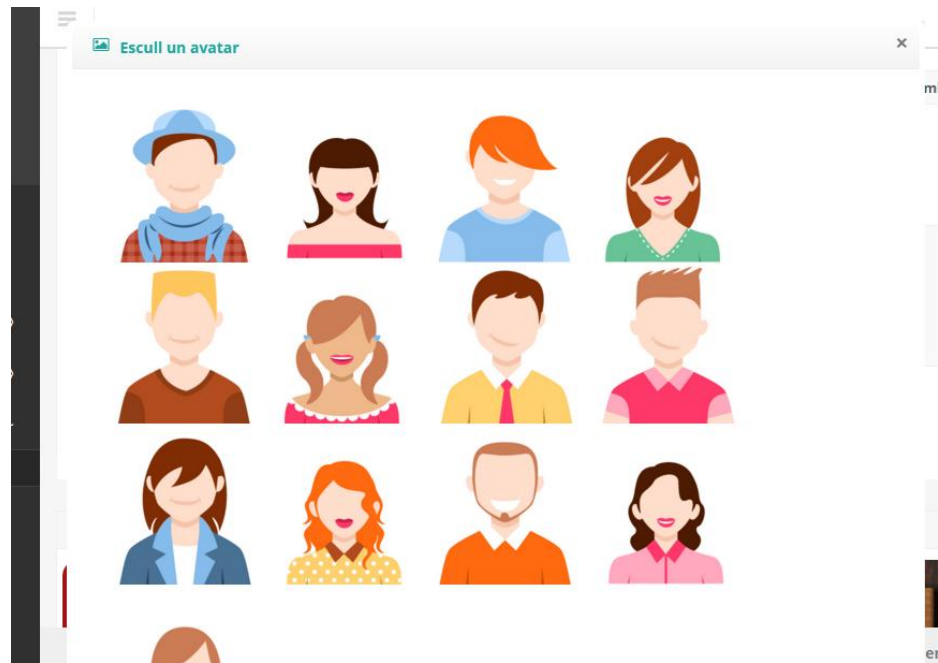
II-lustració 36 - Menú superior dret



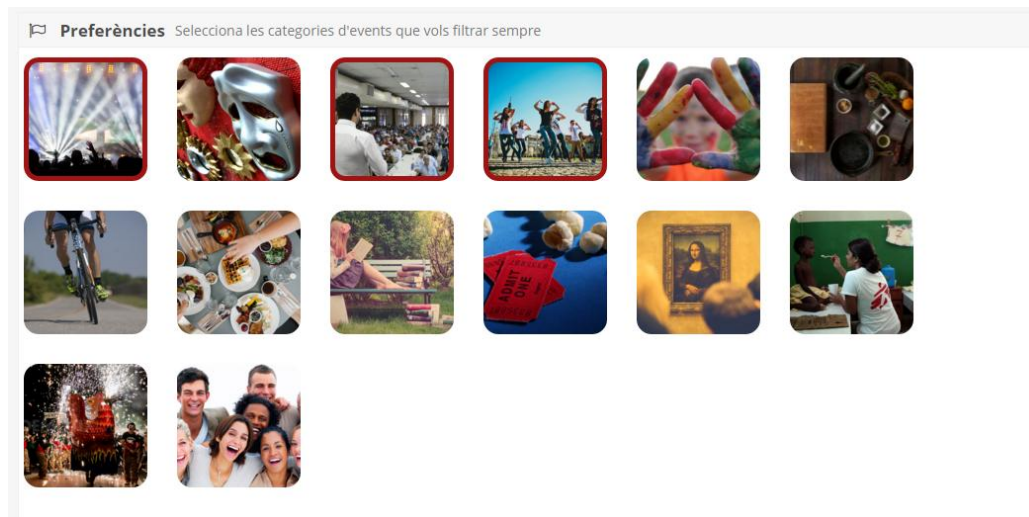
II-lustració 37 - Perfil del propi usuari



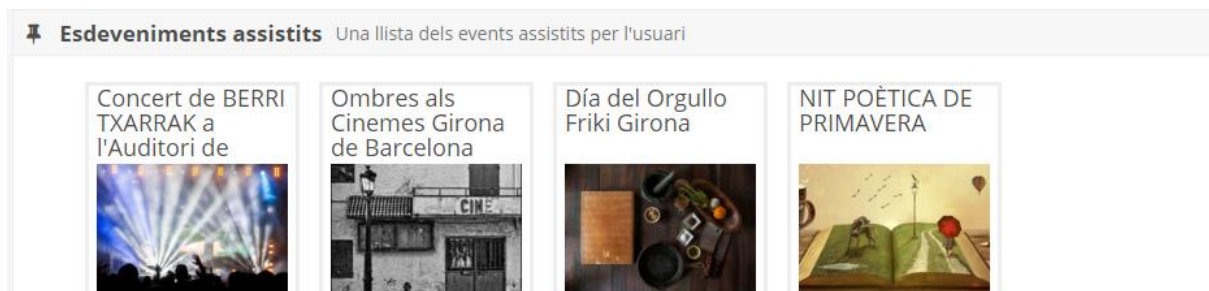
II-lustració 38 - Part del perfil que l'usuari pot omplenar o triar avatar



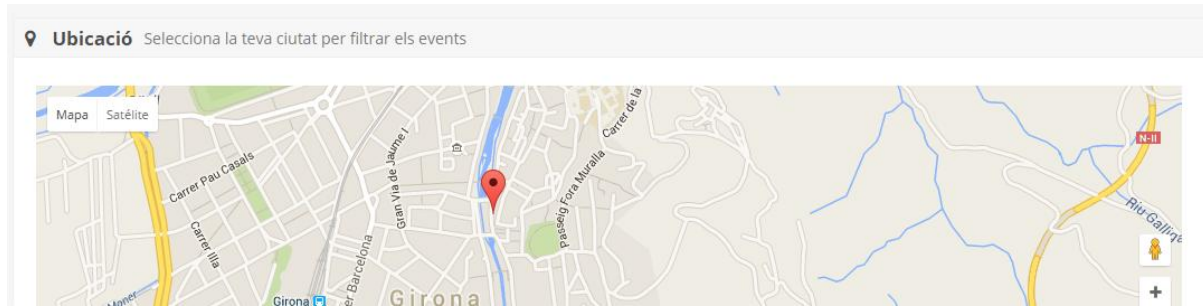
Il·lustració 39 - Popup on es mostren els avatars disponibles



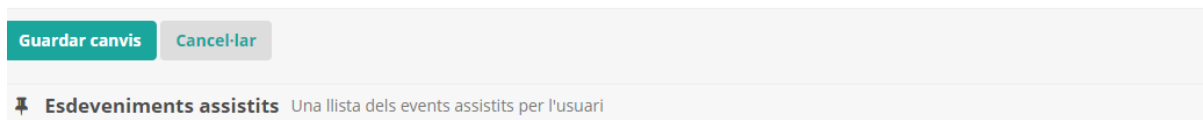
Il·lustració 40 - L'Usuari pot triar tot prement les icones de les categories



Il·lustració 41 - Llistat d'esdeveniments assistits per l'usuari

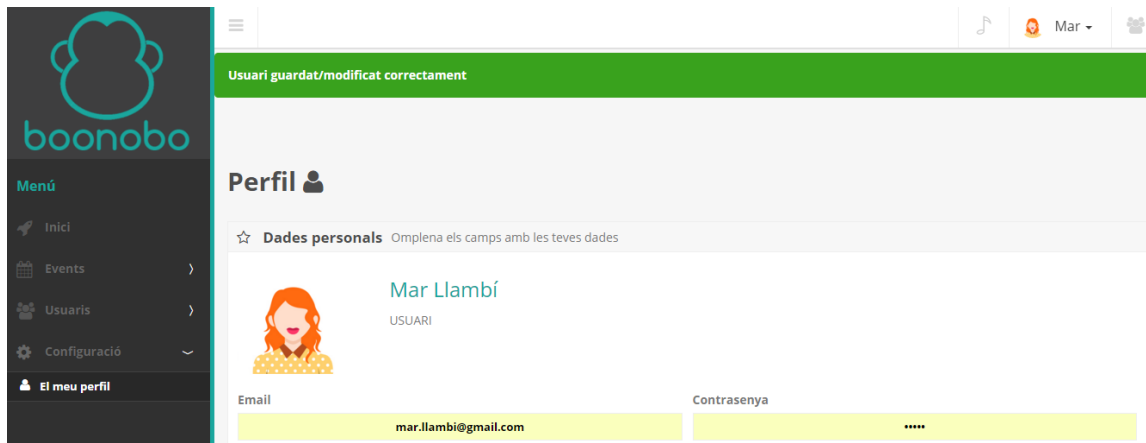


II-lustració 42 - Mapa de l'ubicació de l'usuari



Concert de BERRI | Ombres als | Día del Orgullo | NIT POÈTICA DE

II-lustració 43 - Situació dels botons de guardat o cancel·lar



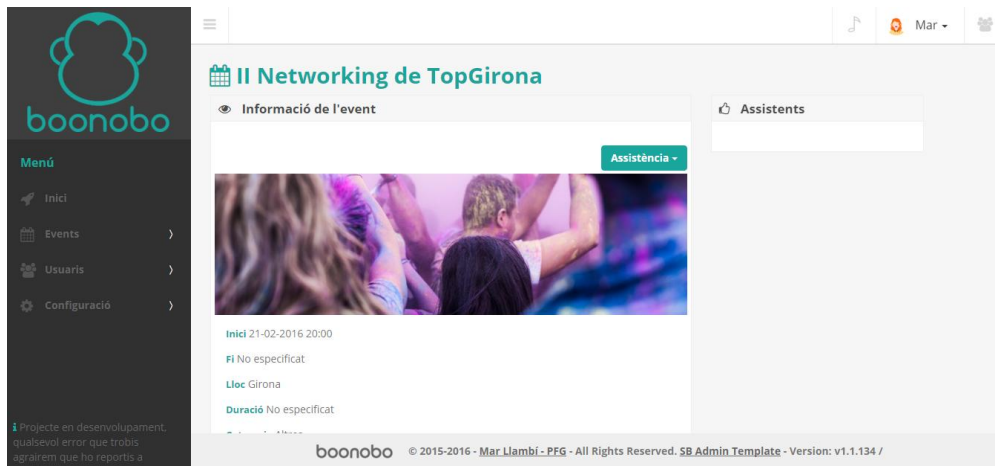
II-lustració 44 - Confirmació dels canvis guardats

### 10.2.11- Visualitzar esdeveniment

Des de qualsevol miniatura d'esdeveniment [Figura 45], que podem trobar en varis apartats de l'aplicatiu, podem accedir a la informació d'aquest. Un cop es prem la miniatura de l'esdeveniment, l'aplicatiu ens redirigirà a la pàgina amb tots els detalls. En el cas que ens trobem a la pàgina de recomanants, es mostra una opció a la part superior de la miniatura per deixar de veure l'esdeveniment en la llista. Un cop estem a la pàgina de l'esdeveniment, podem veure tota la informació (inclòs la llista d'assistents, si n'hi ha). A la banda dreta superior a la imatge del esdeveniment podem canviar la nostra relació amb l'esdeveniment: M'agradaria, Assistiré o no Assistiré [Figura 46-47]. Un cop canviem l'estat, es queda l'opció marcada per l'usuari.



II-lustració 45 - Miniatura d'un esdeveniment



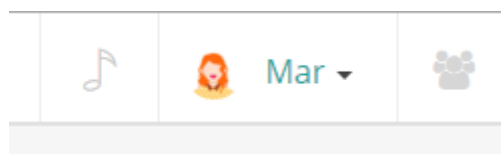
II-lustració 46- Perfil o pàgina de l'esdeveniment



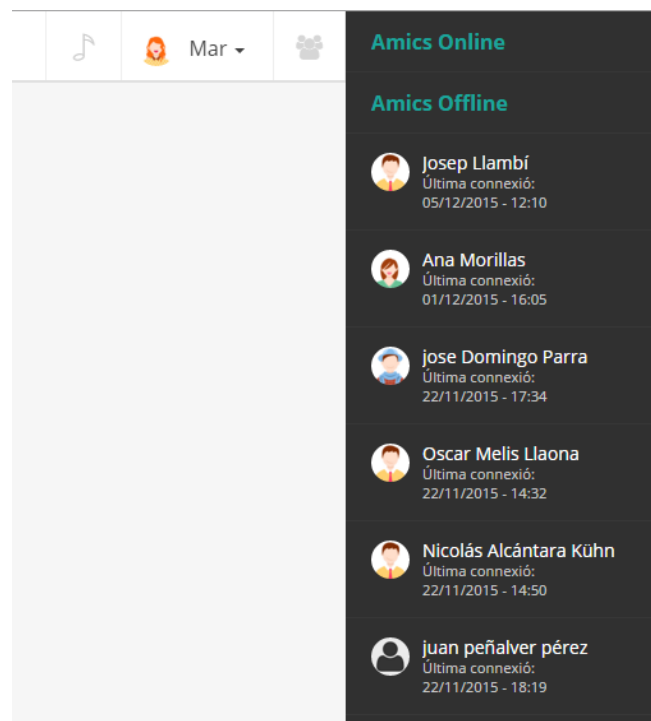
II-lustració 47 - Desplegable per establir una relació entre l'usuari i l'esdeveniment

### 10.2.12- Xatejar amb usuaris

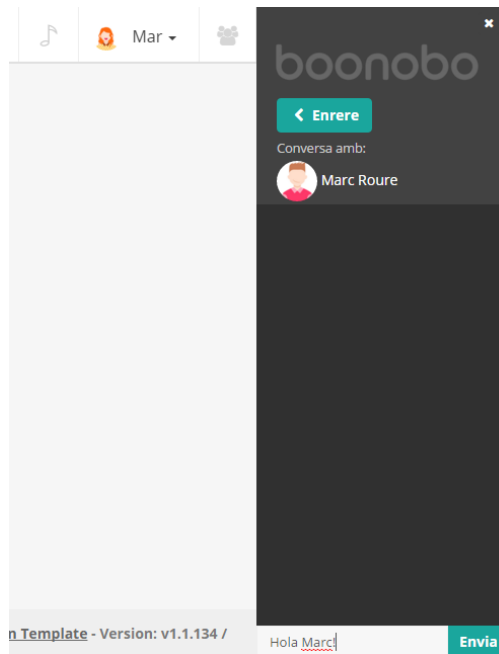
Al menú superior la icona de més a la dreta [Figura 48], que simula un grup de gent, activa el xat d'usuaris. Si es prem aquest botó, es desplega una llista amb tots els usuaris *Online* i *Offline* que estiguin en la nostra llista d'amistats [Figura 49]. Per cada usuari se'ns mostra el seu nom, avatar i última data de connexió. Si es prem un usuari, encara que estigui *Offline*, entrem en la conversa amb aquest usuari [Figura 50]. Podem enviar-li missatges i els llegirà un cop entri en l'aplicatiu [Figura 51]. Si algú ens enviés un missatge, l'aplicatiu ens avisaria amb un missatge en *flash* i a la llista d'usuaris del xat, es veuria una notificació diferent.



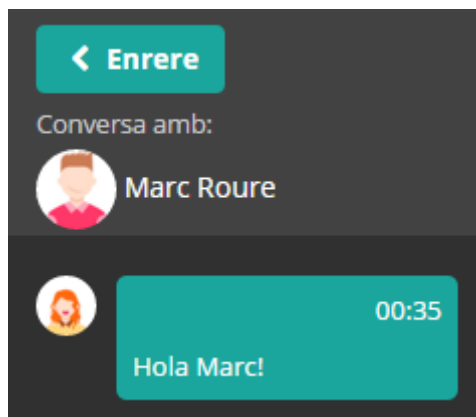
Il·lustració 48 - Menú superior dret



Il·lustració 49 - Desplegable d'usuaris del xat



II-lustració 50 - Conversa amb l'usuari Marc Roure



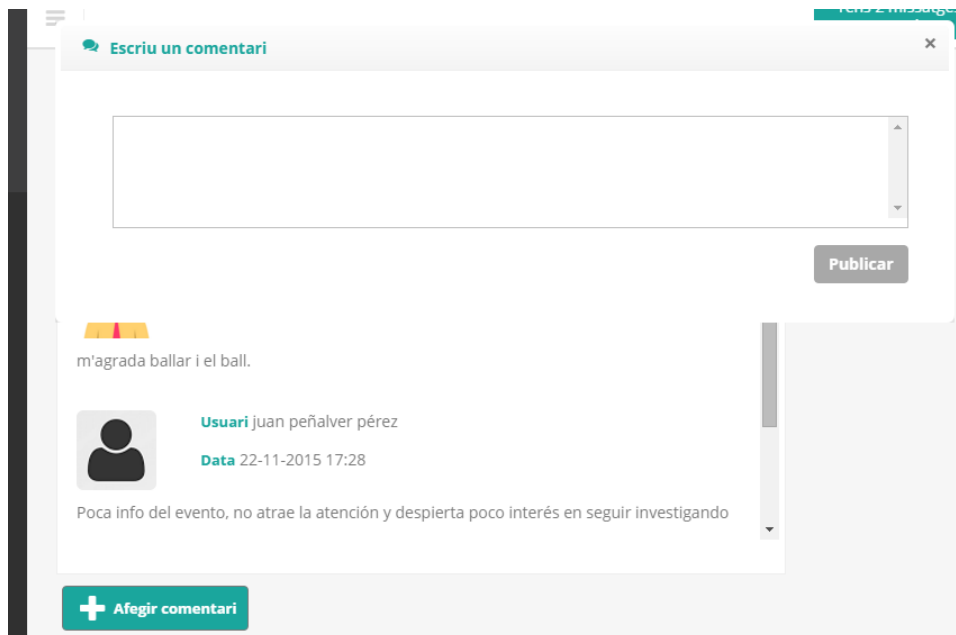
II-lustració 51 - Vista del missatge enviat a l'usuari Marc Roure

### 10.2.13- Afegir comentaris en un esdeveniment

Quan entrem en la pàgina d'un esdeveniment, al final de tota la informació, trobem l'apartat de comentaris [Figura 52]. Allà podem visualitzar una llista de tots els comentaris fets per usuaris i la data de publicació. Just a sota, veiem el botó per afegir un nou comentari; quan el premem, s'obre un *popup* per redactar la publicació [Figura 53]. Podem tancar amb la “x” de la part superior dreta o publicar amb el botó de “Publicar”. Si es publica correctament, es mostra un missatge en verd [Figura 54] al recarregar la pàgina, altrament el missatge serà en vermell. Si baixem a l'apartat de comentaris, ja hi sortirà el nostre publicat.



Il·lustració 52 - Llista de comentaris de l'esdeveniment



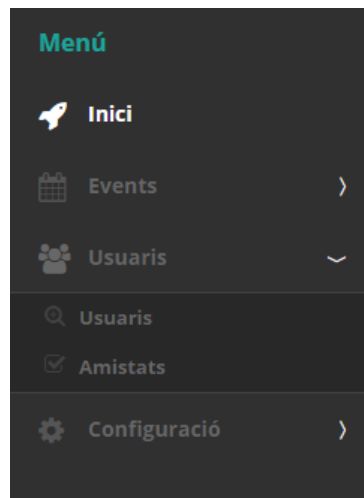
Il·lustració 53 - Popup per escriure un nou comentari



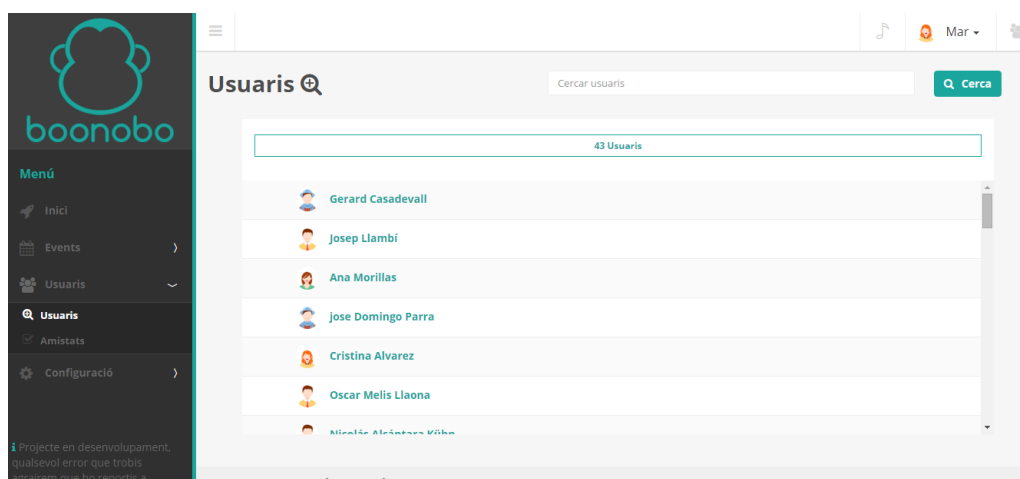
Il·lustració 54 - L'aplicació ens confirma que s'afegeix el comentari

### 10.2.14- Cercar usuaris

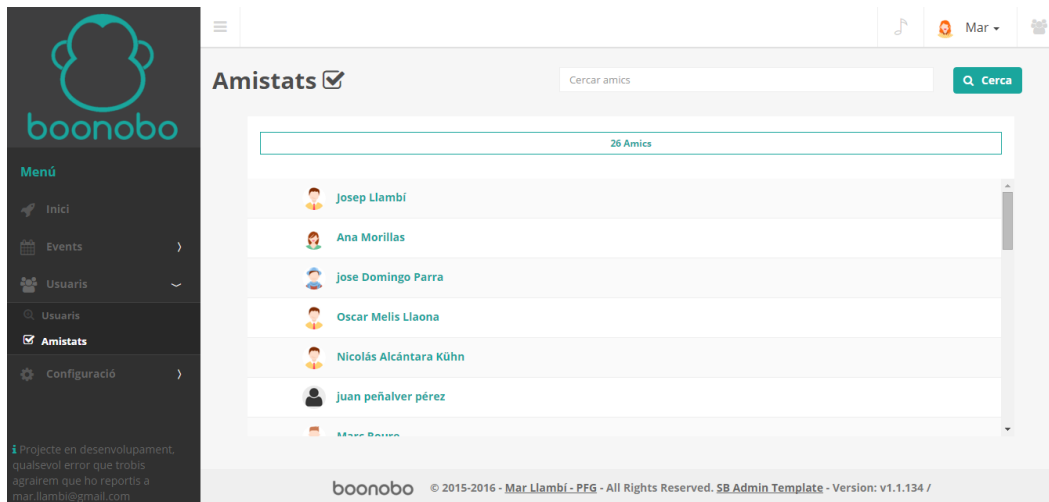
En el menú de mà esquerra trobem l'apartat d'Usuaris, si hi premem, se'ns desplega un submenú amb dos apartats nous [Figura 55]: Usuaris i Amistats, ambdós s'expliquen en aquest apartat perquè tot i que poden semblar diferents, funcionen iguals. Si anem a usuaris, se'ns rediregeix a una nova pàgina [Figura 56] on veiem una llista de tots els usuaris de la xarxa social. A la banda esquerra superior tenim un cercador per buscar per nom o cognom o ambdós a qualsevol usuari. Si anem al subapartat amistats, se'ns redirigeix a una pàgina de característiques semblants [Figura 57] amb la única diferencia que la llista es del conjunt d'usuaris que tenim agregats a la nostra llista d'amistats. Podem comprovar que a la banda superior esquerra també hi és el buscador que realitza la cerca per a qualsevol mot que formi part del nom d'alguna amistat nostra [Figura 58]



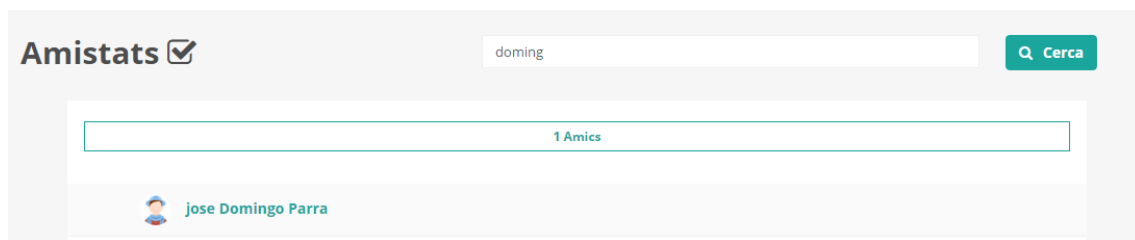
II-lustració 55 - Menú esquerre desplegat en "Usuaris"



II-lustració 56 - Llistat de tots els usuaris de la plataforma



II-lustració 57 - Llistat de totes les amistats



II-lustració 58 - Cercador d'usuaris a través del nom o cognoms

## 11. Conclusions o assoliment dels objectius

A continuació s'exposen totes les conclusions que es deriven d'aquest projecte. S'han classificat en tres apartats diferents: conclusions derivades dels resultats de la implementació (es presenta l'anàlisi de la valoració dels usuaris), conclusions derivades de la implementació de l'aplicatiu i compliment de la legislació vigent.

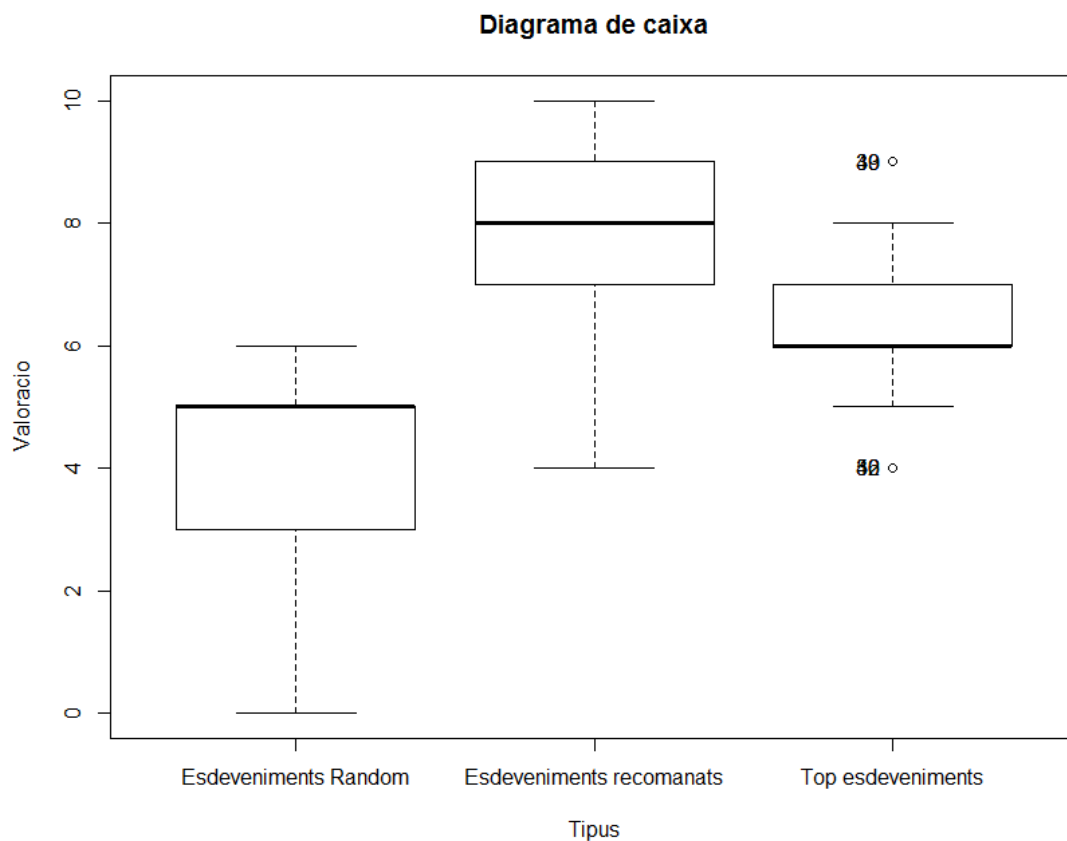
### 11.1 - Conclusions derivades dels resultats

Com s'ha comentat en apartats anteriors, els possibles errors de l'aplicatiu web han estat corregits gràcies al *feedback* dels usuaris. Però, com verifiquem que el sistema recomanador funciona? La única manera de saber si un sistema recomanador funciona realment és treballant amb dades reals. Això és important, perquè no només es pot extreure una conclusió ràpida del funcionament del sistema recomanador, sinó que també dóna les claus per millorar-ne el disseny.

Així doncs, es va demanar a tots els usuaris que s'havien registrat a la pàgina web que emplenessin un formulari on havien de donar tres puntuacions: la qualitat dels esdeveniments dels apartats recomanats, top 20 i *random*. La peculiaritat però, és que cap d'ells sabia que en realitat el contingut que es mostrava (la llista d'esdeveniments) estava creuada. Què significa això? Es van separar internament els usuaris en tres grups, cadascun d'ells, quan accedís a un dels tres apartats d'esdeveniments de la web, se li mostraria un conjunt diferent d'esdeveniments. De manera que un dels grups d'usuaris, quan anaven a esdeveniments Recomanats apareixien esdeveniments del Top 20, quan anaven a Top 20 es mostrava la llista d'esdeveniments Random i per últim, si anaven a esdeveniments Random se'ls mostrava els esdeveniments recomanats. Això es realitzava totalment en background de manera que de cares a l'usuari, no notava la diferència i visitava els apartats convençut de llegir la llista d'esdeveniments que hi pertocava.

S'aconsegueix a més que la seva valoració fos més fiable i no estigués condicionada per l'apartat on es trobaven (per aquest motiu se'ls hi va demanar rigor a l'hora de puntuar).

Un total de 27 usuaris van respondre el formulari i les respostes es van emmagatzemar automàticament en un *excel*. Es va exportar l'*excel* en format *.csv* a fi de poder treballar amb les puntuacions de manera estadística amb el programa R Commander.

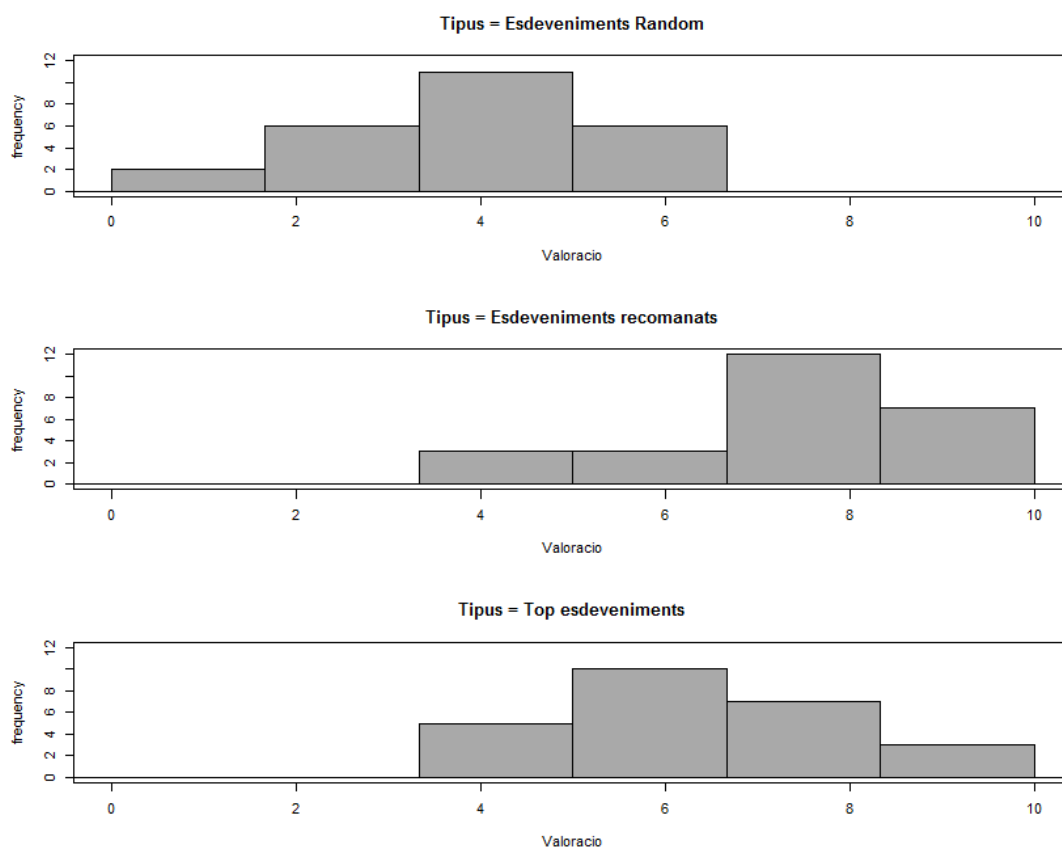


**Il·lustració 59 - Diagrama de caixa de les valoracions dels usuaris**

Amb el diagrama de caixa destacarem primer els valors atípics que s'observen al top esdeveniments fruits del problema que venim plantejant durant tot el projecte: hi haurà usuaris amb qui plenament s'encerti la recomanació d'aquells esdeveniments amb més assistents i al contrari, usuaris que no hi estaran gens d'acord.

Després també observem que els valors mínims als quals arriben les valoracions dels esdeveniments *random* són molt diferents als valors mínims de les altres valoracions.

és curiós observar com les valoracions dels esdeveniments recomanats són simètriques, a diferència dels esdeveniments *random* on observem una clara dispersió en els valor que van del 50% al 25%. En quant al top 20 d'esdeveniments, tindríem el cas oposat, segurament influenciat per casos atípics..



**II-lustració 60 - Histograma resultant de les valoracions dels usuaris**

Quan visualitzem el histograma, ràpidament identifiquem en els tres tipus quines són les freqüències més altes. Novament observem una tendència cap a l'esquerra de les valoracions dels esdeveniments *random* i, en aquest cas, podem veure que els altres dos tipus comparteixen tendència cap a la dreta.

A diferència del primer i el tercer gràfic que tenen una certa homogeneïtat entre costats, el segon presenta una clara diferència entre la freqüència mitjana que aniria en torn els 8, amb els mínims com seria entorn el 4.

Per acabar, analitzarem els contrastos de mitjana formulant les hipòtesis per cada conjunt de mitjanes dels tres tipus d'esdeveniments. Suposem tenim dues mitjanes  $X_1$  i  $X_2$  mostrals d'una població. Volem saber si la diferència de les mitjanes és significativa o no, és a dir, si podem acceptar que  $\mu_1 = \mu_2$

En el primer cas suposem  $X_1$  és la mitjana d'un subconjunt de valoracions d'esdeveniments Random i  $X_2$  la mitjana d'un subconjunt de valoracions d'esdeveniments recomanats. Plantegem les següents hipòtesis

- Hipòtesi nul·la  $\rightarrow H_0 : \mu_1 - \mu_2 = 0$ , és a dir,  $\mu_1 = \mu_2$ ,
- Hipòtesi alternativa  $\rightarrow H_a : \mu_1 - \mu_2 < 0$ , és a dir,  $\mu_1 < \mu_2$

#### Paired t-test

```
data: Esdeveniments.Random and Esdeveniments.recomanats
t = -6.6524, df = 24, p-value = 3.505e-07
alternative hypothesis: true difference in means is less than 0
95 percent confidence interval:
 -Inf -2.466157
sample estimates:
mean of the differences
 -3.32
```

Observem que després d'executar un test-t per a mostres relacionades, ens retorna un valor p de 3.5e-07. Si acceptem p-valors a partir de 0.05, podem dir que rebutgem la hipòtesi nul·la i ens quedem amb l'alternativa: que si aquestes mostres amb les que treballem representessin un subconjunt d'una mostra més gran, la mitjana de valoracions pels esdeveniments Random seria més petita que la d'esdeveniments Recomanats.

#### Paired t-test

```
data: Top.20.esdeveniments and Esdeveniments.recomanats
t = -2.0079, df = 24, p-value = 0.02802
alternative hypothesis: true difference in means is less than 0
95 percent confidence interval:
 -Inf -0.1538578
sample estimates:
mean of the differences
 -1.04
```

En aquest segon cas situem amb la mateixa hipòtesi nul·la i alternativa dos subconjunts diferents: els extrems de les valoracions del Top 20 d'esdeveniments i altre cop, els esdeveniments recomanats. A l'executar el t-test, obtenim un p-valor de 0,02 que altre cop ens descarta la hipòtesi nul·la i acceptar que si aquestes dades són mostres d'un conjunt, la

mitjana d'esdeveniments del Top 20 serà inferior a la d'esdeveniments Recomanats.

```

Paired t-test

data: Esdeveniments.Random and Top.20.esdeveniments
t = -6.5348, df = 24, p-value = 4.645e-07
alternative hypothesis: true difference in means is less than 0
95 percent confidence interval:
 -Inf -1.683068
sample estimates:
mean of the differences
 -2.28

```

I si analitzem, sota les mateixes condicions, les valoracions dels esdeveniments Random i el Top 20 d'esdeveniments? Doncs que novament amb una probabilitat tant petita com és 4.64e-07 descartem la hipòtesi nul·la i ens quedem amb l'alternativa: que la mitjana d'esdeveniments Random serà inferior a la d'esdeveniments Recomanats.

Després d'aquests anàlisis estadístics podem extreure dues conclusions globals: Per començar que la de valoracions d'esdeveniments Recomanats sempre serà superior a la d'esdeveniments Random i esdeveniments Top 20 i que, d'altra banda, la raó del sistema col·laboratiu està reforçada amb la conclusió que la mitjana de valoracions d'esdeveniments Top 20 serà superior a la mitjana de valoracions d'esdeveniments Random.

Aquestes conclusions reforcen el treball realitzat en aquest projecte i acaben reforçant els propòsits tècnics complerts amb els propòsit teòric que el sistema recomandor funcionés millor que mostrar esdeveniments aleatoris sense raó.

Cal estudiar però tots els casos i trobar relacions entre els usuaris que han puntuat més alt o més baix els tres sistemes i quin han estat les categories o els esdeveniments concrets que ho han influenciat.

Després, amb la base de dades, ens preguntem quin han estat els esdeveniments més recomanats en el moment de redacció d'aquestes línies, perquè hi són i si deuen haver influït en aquestes valoracions. Després d'una simple consulta obtenim que els 5 esdeveniments més recomanats són:

Id esdeveniment	Total usuaris	Categoria de l'esdeveniment
429	35	Cinema
4489	34	Teatre i dansa
1883	34	Concert
600	34	Infantil
318	34	Concert

Podem veure que no hi ha cap categoria que predomini en aquest Top 5 de recomanacions. Es podrien mirar els motius, els perfils d'assistents a aquests esdeveniments i els perfils dels usuaris els quals se'ls hi ha recomanat (podríem trobar subconjunts d'usuaris dins l'aplicació).

Quina es correspon amb la categoria d'esdeveniment més recomanada? Fent un *Join* amb la consulta utilitzada per obtenir la taula anterior i afegint la informació de la taula d'esdeveniments veiem que aquestes són les 5 categories més recomanades:

Categoria	Total recomanacions
Concert	332
Cinema	288
Cursos i tallers	282
Teatre	176
Conferències i presentacions	112

Suposem però, que els esdeveniments estan ben classificats i es corresponen amb la categoria que apunten. A partir d'aquí podríem realitzar comparatives de si les categories més recomanades es corresponen amb les que més assistents tenen i veure a quines categories corresponen els esdeveniments que s'han recomanat als usuaris que millor han

puntuat el sistema recomanador. Tot un conjunt d'estadístiques que ens ajudaria a entendre el comportament dels usuaris i millorar l'efectivitat del sistema recomanador de cares a un treball futur.

## 11.2 - Conclusions derivades de la implementació

Les conclusions que aquí s'exposen seran enumerades segons els objectius i motivacions esmentades a l'apartat 1. S'afegeixen també conclusions extretes durant la realització d'aquest projecte, diferents als objectius marcats, i durant la redacció d'aquesta memòria.

### **Ampliar el coneixement sobre Sistemes Recomanadors**

Aquest projecte m'ha proporcionat la oportunitat d'endinsar-me en el la branca de la intel·ligència artificial que més interès em despertava. He estat capaç de realitzar un curs sencer *online* sobre introducció a aquests sistemes i assolir els coneixements bàsics per planificar-ne un projecte.

### **Aprenentatge de noves tecnologies i metodologies**

Durant aquest projecte no només s'ha aconseguit més coneixements teòrics sinó que també s'ha millorat les habilitats ens molts aspectes tècnics.

- Del llenguatge Java que ja coneixem, s'ha après a utilitzar noves llibreries i millorar la configuració.
- Tot i que es tenia coneixement de cakePHP, ha estat amb aquest projecte com s'ha assolit un coneixement global i s'ha entès el funcionament de bootstrap per aconseguir webs *responsive*.
- Els llenguatges PHP i Javascript han estat els més destacables en quan a aprenentatge se'n deriva, llenguatges que normalment utilitzem per desenvolupar les mateixes funcions, aquest cop s'ha anat més enllà.
- Tot el funcionament d'Amazon Web Services i la configuració per consola d'un servidor. Per aconseguir-ho, abans han calgut moltes hores de tutorials i de recerca

### **Entendre l'ètica sobre la captura de dades implícites de l'usuari**

És curiós però, la privacitat sempre necessita de l'ètica en el moment en que tractem qualsevol dada que prové d'un usuari. Qualsevol perfil d'usuari elaborat a partir de les

costums d'aquest pot aproximar als altres usuaris a descobrir de quin usuari en qüestió es tracta. Quan més precís sigui el perfil que elaborem, més estarem exposant dades que posin en perill la privacitat del nostre usuari. El gran repte que s'obra dins aquest món és el de ser capaços de crear sistemes automatitzats i auto-aprenents que tinguin uns límits en quant a la recol·lecció de les dades es tracta. “Amb l'ètica hem topat” diuen els grans experts en aquesta àrea i és que, si algú pensava que la única limitació és la purament científica, és que no ha entès l'essència dels sistemes recomanadors. Ha estat curiós entendre aquest món i consultar la legislació europea que es vol fer al respecte.

### **Les limitacions dels sistemes recomanadors**

Els sistemes recomanadors són una branca de la intel·ligència artificial que, encara que no ho sembli, és la menys desenvolupada i on menys investigació s'hi ha dedicat. El motius són els grans reptes que es plantegen com l'escassetat de dades, i és que els sistemes recomanadors comercials es basen en grans conjunts de dades que s'enfronten al gran problema analític de la dispersió d'aquestes dades; també l'engegada en fred d'un sistema recomanador posa sobre la taula diferents tècniques, cap encara donada com a vàlida, en la posada en marxa o el primer contacte amb aquests sistemes. Un altre problema és la introducció d'un nou element en una gran base de dades on molts usuaris ja han estat interaccionant amb tots els ítems i aquest element queda descompensat.

I què passa si les dades creixen massa i el nostre sistema recomanador no està preparat? Continuarà funcionant igual de bé? I el tractament de dades es donarà amb el mateix temps (recordem que com a mínim és de  $O(n^2)$ ).

En el nostre cas, el sistema també es veuria afectat pel que s'anomena “Sinonimia” un dels altres grans problemes d'aquesta ciència, i és que un sistema, per molt intel·ligent que el dotem, mai serà capaç de reconèixer dos esdeveniments iguals: pel que “pel·lícula infantil” no serà mai igual que “projecció per a nens”.

Un altre repte que va posar sobre la taula el gegant Amazon són els anomenats usuaris “ovella gris”, aquells que es difícil recomanar quelcom ja que el filtratge col·laboratiu no funciona amb aquests; i després estan els usuaris “ovella negra” els quals és casi impossible fer una recomanació encertada.

I és que tant la diversitat d'opinions, com ara identificar als usuaris que només es dediquen a puntuar negativament, són altres exemples dels reptes actuals que es planteja qualsevol aplicatiu que al darrera hi tingui un SR.

### **El desenvolupament d'una xarxa social per poder interaccionar amb l'usuari**

S'ha aconseguit l'objectiu de desenvolupar un aplicatiu web amb totes les característiques de xarxa social per recopilar informació i construir així una interfície on es mostri també l'*output* del sistema. S'ha requerit d'una planificació específica i sí que és veritat que, a diferència del Sistema Recomanador, ha portat més inconvenients dels esperats. És normal que el *frontend* sempre acabi donant algun error que costi de solucionar i al final això és el que ha tret més hores; era però, essencial pel projecte que funcionés adequadament.

### **La implementació d'un Sistema recomanador a partir de les dades escollides**

Puc dir amb molta satisfacció, que aquest era un dels dos grans pilars d'aquest projecte i estic molt satisfeta amb la feina realitzada. No pel propi funcionament del sistema -pot anar millor o pitjor- sinó perquè des d'un bon inici no m'hauria arribat a imaginar la quantitat d'informació que he arribat a aprendre. Aquest projecte és, sense cap mena de dubte, el primer esgraó en aquesta especialitat i el que m'ha fet obrir els ulls a tot una disciplina nova. La importància principal d'aquest projecte no era fer un sistema recomanador perfecte, que si era important però no principal, sinó assentar les bases d'un coneixement per a un treball futur amb un projecte des d'on partir.

### **La recol·lecció de les opinions dels usuaris**

Les proves del funcionament d'un aplicatiu es fan a partir de test automàtics i test d'usuaris. Així, se'n verifica el seu correcte funcionament i es detecta els possibles errors que puguin haver-hi. Ha estat una gran experiència poder tenir el *feedback* directe dels usuaris en quant a l'aplicatiu web es refereix. M'ha permès gestionar les incidències amb la feina que hi havia per fer i trobar els meus propis errors. Però, i el sistema recomanador? es tractava d'un software que només el podien validar aquells usuaris els quals se'ls hi havien recopilat les dades. Per suposat l'avaluació és subjectiva però, com he pogut comprovar en l'anàlisi de les dades, ajuda al programador a fer-se una opinió d'allò que desenvolupa. És una pràctica poc convencional en un projecte de final de grau, tenir tot un procés de test real com aquest, i m'ha pogut aproximar, a petita escala, al que seria en un projecte gran per empresa.

### **La implementació d'un sistema que es nodreixi d' ElasticSearch**

Aquest era un dels objectius inicials que es volia desenvolupar en aquest projecte. Si bé s'ha explicat en apartats anteriors que la seva raó de ser-hi perdia força per el canvi d'enfocament que s'havia donat al treball, tampoc hauria estat descabellat incorporar-lo. El filtratge per contingut hauria estat més complex encara que no hagués funcionat 100% amb tots els esdeveniments. L'altre motiu de no incloure-ho, que s'explica en aquestes línies, va

ser un motiu personal. Per indisposició mèdica vaig perdre el temps necessari per desenvolupar-ho i incloure-ho al projecte i després de fer una valoració de la importància del *backlog*, es va descartar.

### **Planificar i desenvolupar un projecte informàtic**

M'agradaria fer un apunt: tot i que aquest projecte hagi estat desenvolupat dins el marc d'una *startup*, la planificació de tot ell ha estat elaborada per l'autora del projecte.

Se'ns explica a la carrera que, un bon enginyer no és aquell que sap desenvolupar millor que un altre les habilitats tècniques (com podrien ser programació), sinó aquell que és capaç de tenir els coneixements suficients per planificar un projecte segons els seus requeriments. Què significa això? identificar quines tasques necessitaran més temps que altres no per la quantitat de feina, sinó per la complexitat, resoldre conflictes de calendari quan quedin tasques per fer i no s'arribi als terminis, ser capaç de redactar una memòria on quedin reflectits tots els mesos de treball... un seguit de tasques que al principi sembla que ens hagin d'anar grosses però si un té la confiança suficient amb el què ha après, se'n pot sortir perfectament. Durant aquests mesos he hagut de treballar amb clients que tant podien ser els meus usuaris com els socis de la meva *startup*; He hagut de prendre decisions sobre un sistema (com és el SR) on fa 6 mesos enrere no tenia ni una vuitena part dels coneixements que tinc ara. Una mirada enrere que em fa sentir satisfeta de com he planificat, desenvolupat i reaccionat davant les adversitats, en la realització d'aquest projecte: el meu projecte de final de grau.

## **11.3 - Compliment de la legislació vigent.**

En aquest projecte, com és d'esperar al realitzar perfils d'usuari, és molt important seguir la legislació vigent sobretot amb les dades personals. Cal però, recordar que es tracta d'un projecte el qual l'aplicatiu web no s'utilitzarà, en principi, pel treball futur. Per tant serà la *startup* que hi ha al darrera que, abans de publicar el projecte definitiu, realitzi tots els tràmits per registrar els fitxers de seguretat, designi un responsable de seguretat i tingui cura de la qualitat de les dades.

Començant per la LOPD, el que sí es fa en aquest projecte és un pas previ a això, com és informar a l'usuari de la recollida de dades i que sense el consentiment d'aquest no es pugui dur a terme el registre. Aquest pas és l'acceptació dels termes i condicions on s'informa a l'usuari que es recollirà tot tipus d'informació del seu comportament vers la pàgina i les següents dades personals: Email, Nom i Cognoms, Contrasenya encriptada i Ciutat. A més

d'altre tipus d'informació lliure a omplir per l'usuari com és la professió i descripció personal. La contrasenya tot i que no ens diu res de l'usuari, cal explicar que es guarda i com es fa (encriptada), per possibles accessos aliens a la base de dades. El nom, *email* i ciutat són considerades dades de nivell bàsic, per tant el servidor on estan allotjades ha de cobrir les mesures bàsiques de seguretat. Amazon Web Services compleix tots els requisits necessaris per satisfer la qualitat i la seguretat d'aquests tipus de dades en els seus servidors; tal i com indiquen en el seu sistema, és alta seguretat, amb sistemes de vigilància electrònica, sistemes de control d'accés i guardes de seguretat les 24 hores del dia.

Una altra llei que s'havia de complir era la LSSI, concretament l'article 22.2 (comunament anomenat llei de *cookies*) on es regula l'ús de *cookies* i d'informació emmagatzemada de l'usuari; només després d'haver facilitat informació a l'usuari, avisat i comptar amb el consentiment d'aquest, es podrà dur a terme la recuperació de dades de l'usuari. S'ha afegit un avís en tot moment a la pàgina web, amb el que s'estableix la llei, on es pot accedir a més informació, tancar el missatge (donant a entendre que ho accepta) o continuar navegant sabent què suposa per a l'usuari.

## 12. Treball futur tecnologies o aplicacions existents.

Les característiques en les que s'ha desenvolupat aquest projecte, permeten que ens imaginem un treball futur per aquest. Boonobo és l'aplicatiu principal del qual parteix aquest projecte de final de grau, aquest és la raó d'inici del sistema recomanador i on s'incorpora un cop finalitzada la seva entrega.

Per tant podem dir que el futur del sistema recomanador està assegurant, en tant que se pressuposa que el de Boonobo ja ho està. El gran avantatge d'aquest projecte és que ha estat provat ja pels usuaris, compta amb una comunitat d'aquests i amb un data set d'entrada d'esdeveniments que s'actualitzarà en un futur per seguir mantenint l'aplicació de prova nodrida. És una gran punt a favor d'aquest projecte, ja que són molts, que acaben sent desenvolupats i testats pel desenvolupador, però no compten amb tota una gestió d'incidències i el feedback real de l'usuari. El sistema recomanador ha estat dissenyat, analitzat, desenvolupat, testat, arreglat i posat a prova pels usuaris; un cicle que poques aplicacions d'aquestes característiques poden complir i la justificació dels seus resultats. Era molt important per a mi, obrir i tancar aquest cicle, encara que fos bàsic, per poder tenir una visió real de totes les fases de desenvolupament del software. Queden però, moltes qüestions a tractar que s'han quedat en el tinter i que a continuació resumiré:

- En el recull de feedback dels usuaris, van ser moltes propostes i opinions del disseny de la pàgina web que es van recollir. No es van dur a terme perquè l'aplicatiu no era l'eix central del projecte, això no obstant es tindran en compte pel disseny definitiu de Boonobo.
- A l'aplicatiu final sí que estarà desenvolupat un sistema Elastic Search pel que caldrà trobar una manera d'emplenar tots els camps buits (*nulls*) de les taules de la base de dades dels esdeveniments per tal d'optimitzar el seu ús.
- El sistema recomanador caldrà millorar-lo amb noves funcionalitats. Potenciar el filtratge per contingut amb les característiques que obtinguem de l'ElasticSearch i el seu More Like This per millorar els resultats. També caldrà treballar amb la tria dels veïns de cada usuari, i aquí és on entra el càlcul de similituds entre usuaris.
- A partir de la millora del càlcul de similituds entre usuaris, implementar un sistema recomanador d'usuaris que es basi en aquesta característica, entre altres.
- Ara mateix el sistema recomanador és executat per una consola en *background* i un *crontab*. Caldrà veure, no obstant, com canvia això i com canvia l'execució del sistema recomanador amb una plataforma tan gran: no s'hauran de recórrer tots els usuaris, sinó potenciar noves recomanacions.
- Resoldre els principals problemes del sistema recomanador com a tal, que s'han descrit en les conclusions: ovella grisa, ovella negra, inici en fred...
- Implementar nous estats entre l'usuari i l'esdeveniment depenent de les complexitats que ofereixi el nou aplicatiu

## 13. Bibliografia

- [1] Ricci F., Rokach L., Shapira B., Kantor B. *Recommender Systems Handbook* (2a ed). Nova York: Springer.
- [2] Jannach D., Zanker M., Felfernig A., Friedrich G. *Recommender Systems: An Introduction* (2011). Nova York: Cambridge University Press
- [3] Sun Z., Han L., Huang W., Wang X., Zeng X.(2014) *Recommender systems based on social networks*. Publicació científica. Xina: College of Computer and Information, Hohai University, Nanjing.
- [4] Ekstrand M. , Riedl J., Konstant J. (2011) *Collaborative Filtering Recommender Systems*. Publicació científica. Now, the essence of knowledge
- [5] Herlocker J., Konstant J., Borchers A., Riedl J.(2014) *An Algorithmic Framework for Performing Collaborative Filtering*. Publicació científica. University of Minnesota: Department of Computer Science.
- [6] Rana C.(2012) *Building a Book Recommender System using time based content filtering*. Publicació científica. India: National Institute of Technology
- [7] Matos V. (2006) *Sistemas de recomendación y personalización*. Universidad Nacional del Sur: minería de la web.
- [8] Konstant J., Ekstrand M. (2015) *Introduction to Recommender Systems*. Curs online de Coursera. University of Minnesota <https://www.coursera.org/learn/recommender-systems/>
- [9] *Fundación Big Data: Análisis predictivo* (2013). Recuperat de : <http://fundacionbigdata.org/analisis-predictivo/>
- [10] Aldiab M. (2015) *Public Cloud War: AWS vs Azure vs Google*. Cloud Academy Blog <http://cloudacademy.com/blog/public-cloud-war-aws-vs-azure-vs-google/>
- [11] Amazon Web Service Web (2016): *las ventajas del servicio*<https://aws.amazon.com/es/application-hosting/benefits/>
- [12] Sánchez Eladio J. (2007) *Comparativa de frameworks en PHP: CakePHP, Symfony y Zend Framework*. Universidad Pontífica católica de Perú: <http://tuxpuc.pucp.edu.pe/articulo/comparativa-de-frameworks-en-php-cakephp-symfony-y-zend-framework>
- [13] Garrido A. (2013) *Comparativa de frameworks*. Universidad de Salamanca: <http://es.scribd.com/doc/189203494/Comparativa-Framework#scribd>
- [14] Tuero L (2013). *Las 12 ventajas del HTML5*. Go Movil: <http://www.go-movil.es/index.php/12-ventajas-del-html5/>


- [15] Luiggi S.M. (2014) *12 ventajas de la programación PHP que deberías saber*. Staff creativa: <http://www.staffcreativa.pe/blog/ventajas-programacion-php/>
- [16] Prezzo M.P (2015). *6 buenos motivos para trabajar con PHP*. Lance Talent Blog: <https://www.lancetalent.com/blog/6-buenos-motivos-para-trabajar-con-php/>
- [17] PHP Página oficial: *¿qué es el PHP?* (2016) <http://php.net/manual/es/intro-what-is.php>
- [18] *Las estrategias del analisis predictivo*. Lantares Solutions Blog. <http://www.lantares.com/blog/las-estrategias-del-analisis-predictivo-y-sus-ventajas>
- [19] García M. (2012) *10 ventajas de utilizar Modelo-Vista-Controlador (MVC)*. Blog de Web In <http://blogdewebin.blogspot.com.es/2012/04/10-ventajas-de-utilizar-model-vista.html>

## 14. Annexos

A continuació s'inclouen tres annexos que no formen part del desenvolupament de l'aplicació en sí però han estat presents en la realització d'aquest PFG.

### 14.1 - Annex 1: document enviat als futurs usuaris per testar l'aplicació

El text que s'adjunta a continuació va ser l'enviat a tots aquells que van mostrar interès en el projecte i volien col·laborar-hi.



**Projecte Boonobo – Sistema recomanador**

Hola! Gràcies per oferir-te/deixar-te convèncer/qualsevol cosa que hagi fet perquè estiguis llegint això ! Et presento el meu Projecte de Final de Grau que consisteix en una plataforma que recomana esdeveniments als usuaris registrats. La teva ajuda consistirà en testejar-la (utilitzar-la una mica) en tres fases diferents i comunicar-me possibles errors que trobis. A la vegada també estaràs generant, sense saber-ho, dades de comportament d'usuari així que et demano que actuis com tu mateix a l'aplicació (no comencis a fer coses sense sentit) perquè la captura sigui el més fiable possible.

Coses a saber abans de començar:

- Estàs firmant un contracte en mi (muajaja) quan facis la fase 1 que es descriu en aquest document, més endavant hauràs de fer la fase 2
- Cada fase te una duració d'uns mínim 15 minuts, màxim... els que tu vulguis!
- La pàgina està en desenvolupament, qualsevol error que trobis hi ha un mail a la pàgina on pots comentar-m'ho. Si no trobes cap, perfecte!
- Els esdeveniments que trobaràs a la web, encara que la data indiqui que es realitzaran en el futur, són esdeveniments capturats fa mínim 5 mesos (pel que et poden semblar.. "antics"). Et demano una mica d'imaginació i si per exemple és una festa major d'estiu, fassis veure que estem a l'estiu etc.

- Si no trobes esdeveniments que t'agradin... perfecte! Fes-m'ho saber també! És important! - Els events són capturats de forma automàtica pel que pot ser que alguns no tinguin tota la informació que ens agradaria. Isn't my fault! Si no t'agrada l'event.. next! Si t'agrada però vols tenir més informació fes el q faries al facebook per exemple: comenta'!!

Fase 1:

- Entra a boonobo.es i registra't (amb compte que si no recordes la contrasenya... no podràs entrar!)

- Ves al teu perfil d'usuari i afegeix camps del teu perfil, preferències... per defecte la ciutat de l'usuari és Girona ;) En una següent versió es podrà canviar

- En un inici no hi hauran esdeveniments recomanats per a tu (fins a la fase 2) així que ves a l'apartat Events->Tots els events i comença a filtrar!

- Entra en els events que et creïn curiositat. A l'event podràs fer-hi comentaris, mirar la llista d'altres usuaris que hi assistiran i el més important: marcar el teu estat. Hi aniries? (Deixa volar la imaginació) T'agradaria assistirhi? (Hi aniries però no n'estàs segur) o senzillament aquell event no hi aniries (perquè no pots o perquè no t'acaba d'agradar). Aquest últim pas és super important!

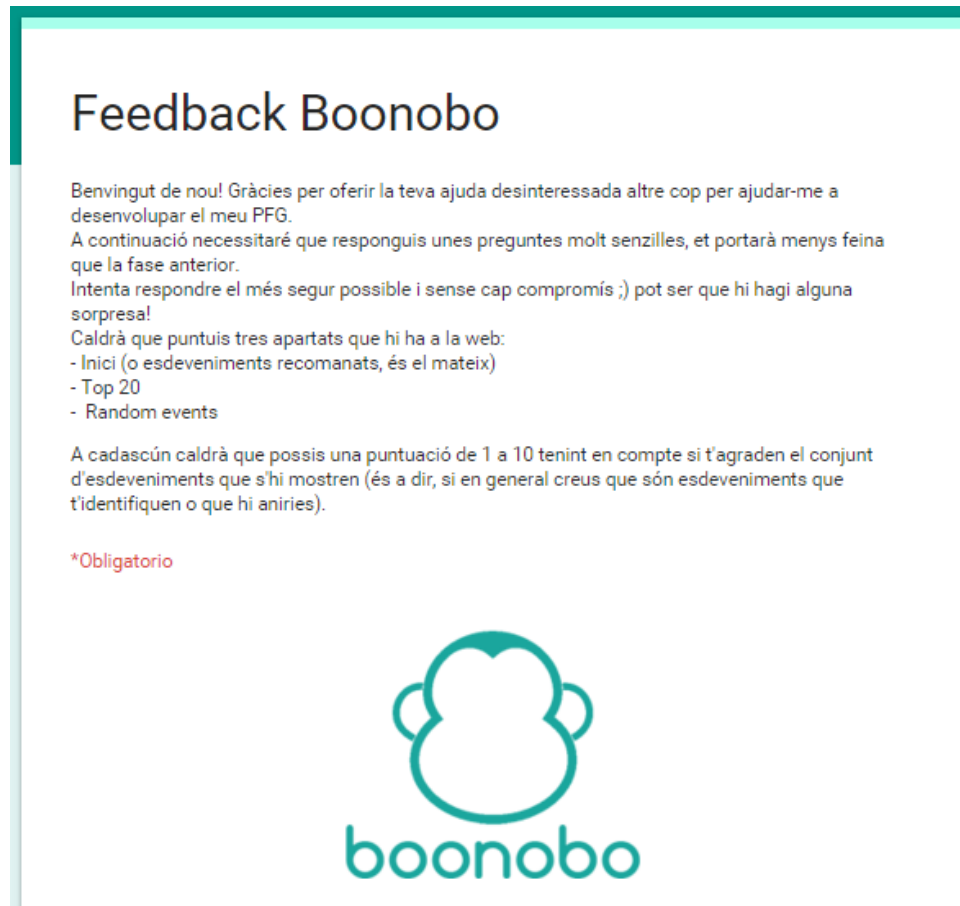
- Per últim ves a l'apartat d'usuaris i fes-te amic d'algun d'ells (potser hi trobes de debò algun conegut) L'amistat es crea automàticament sense peticions.

Fase 2: pròximament em posaré en contacte amb tu perquè la facis

Gràcies! Mar

## 14.2 - Annex 2: formulari google per recol·lectar feedback del sistema recomanador

A continuació s'adjunta el formulari<sup>22</sup> que van respondre els usuaris un cop el sistema recomanador ja estava implementat i el qual ha ajudat a realitzar l'anàlisi final. S'adjunta finalment la taula de puntuacions obtinguda<sup>23</sup>.

The image shows a Google Form titled "Feedback Boonobo". The text inside the form is in Catalan. It welcomes the user and thanks them for their help. It explains that the user will answer simple questions to help improve the PFG system. It asks the user to rate the system on a scale of 1 to 10, considering how much they like the events shown. It also asks the user to rate the events themselves on a scale of 1 to 10. The form includes a list of events: "Inici (o esdeveniments recomanats, és el mateix)", "Top 20", and "Random events". At the bottom, there is a logo for "boonobo" which consists of a stylized cloud-like shape above the word "boonobo".

**Feedback Boonobo**

Benvingut de nou! Gràcies per oferir la teva ajuda desinteressada altre cop per ajudar-me a desenvolupar el meu PFG.

A continuació necessitaré que responguis unes preguntes molt senzilles, et portarà menys feina que la fase anterior.


Intenta respondre el més segur possible i sense cap compromís ;) pot ser que hi hagi alguna sorpresa!

Caldrà que puntuis tres apartats que hi ha a la web:

- Inici (o esdeveniments recomanats, és el mateix)
- Top 20
- Random events

A cadascun caldrà que possis una puntuació de 1 a 10 tenint en compte si t'agraden el conjunt d'esdeveniments que s'hi mostren (és a dir, si en general creus que són esdeveniments que t'identifiquen o que hi aniries).

**\*Obligatorio**



<sup>22</sup> Formulari realitzar a través de Google Groups disponible a:  
<https://docs.google.com/forms/d/10eTvLw3tBVrITZs7BHJIR3ybGvxQEQQovkLxegst4-2o/viewform>

<sup>23</sup> S'han substituït els emails introduïts per l'id de l'usuari a l'aplicació a fi de preservar la privacitat dels usuaris registrats.

El teu mail de registre és...? \*

Tu respuesta

---

Puntuació esdeveniments recomanats \*

0 1 2 3 4 5 6 7 8 9 10

Gens d'acord amb mi ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ Totalment d'acord amb mi

Puntuació Top 20 \*

0 1 2 3 4 5 6 7 8 9 10

Gens d'acord amb mi ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ Totalment d'acord amb mi

Puntuació Random Events \*

0 1 2 3 4 5 6 7 8 9 10

Gens d'acord amb mi ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ Totalment d'acord amb mi

ENVIAR

Puntuació esdeveniments recomanats	Puntuació Top 20	Puntuació Random Events	Id usuari
6	7	3	1
8	7	5	2
8	6	6	3
8	8	4	5
5	9	5	12
9	6	2	13
8	5	5	16
9	6	3	22

10	6	6	24
10	6	6	25
8	4	4	29
9	6	1	36
7	8	3	40
10	9	5	41
5	7	5	42
8	6	6	47
7	4	2	49
8	9	6	54
6	7	5	58
3	6	4	6
7	6	3	18
8	8	5	19
9	5	4	27
8	6	6	32
6	4	0	34

### 14.3 - Annex 3: plannificació de les tasques dels sprints

L'eina utilizada per planificar la feina de desenvolupament ha estat Trello board. Es va descriure un backlog i es van anar assignant les tasques segons prioritat. Les noves tasques que anaven sorgint es van anar assignant als sprints corresponents.

Sprint	Tasques
19/09/2015 - 25/09/2015	<ul style="list-style-type: none"> <li>• Instal·lació del CakePHP en local <ul style="list-style-type: none"> <li>◦ Instal·lació WAMP</li> <li>◦ Creació repositori .git</li> <li>◦ Instal·lació cake</li> <li>◦ Instal·lar plantilla bootstrap</li> <li>◦ revisió instal·lació</li> <li>◦ creació protocol de seguretat i Auth CakePHP</li> <li>◦ Integració Eclipse i Xdebug</li> <li>◦ Creació protocol login</li> </ul> </li> <li>• L'usuari es pot registrar</li> <li>• L'usuari es pot loguejar</li> </ul>
26/09/2015 - 2/10/2015	<ul style="list-style-type: none"> <li>• [Fix]No es guarda bé la timezon de l'usuari</li> <li>• L'usuari pot fer logout</li> <li>• [Fix] No es guarda bé el llenguatge predefinit</li> <li>• Perfil de l'usuari <ul style="list-style-type: none"> <li>◦ L'usuari pot visualitzar el seu perfil</li> <li>◦ L'usuari pot editar el seu perfil</li> <li>◦ L'usuari pot visualitzar el perfil de la resta d'usuaris (amics/no amics)</li> </ul> </li> </ul>
3/10/2015 - 9/10/2015	<ul style="list-style-type: none"> <li>• Pàgina amistats <ul style="list-style-type: none"> <li>◦ Pàgina per visualitzar amistats i buscar per nom</li> <li>◦ Modal per cercar usuaris per nom</li> <li>◦ Modal per afegir usuaris a amistat</li> </ul> </li> <li>• Xat entre usuaris <ul style="list-style-type: none"> <li>◦ L'usuari visualitza els seus amics online/offline</li> <li>◦ L'usuari manté conversa amb un altre usuari</li> <li>◦ Procés per detectar nous missatges online/offline</li> </ul> </li> </ul>

10/10/2015 - 16/10/2015	<ul style="list-style-type: none"><li>● Disseny de la BdD<ul style="list-style-type: none"><li>○ Detectar canvis en la BdD</li><li>○ Passar el nou disseny a workbench MySQL</li><li>○ Omplear taules categories, cities, counries</li><li>○ Crear les relacions user-user d'amistat</li><li>○ Inserir tot el data sets d'esdeveniments</li></ul></li><li>● Neteja data-sets d'esdeveniments<ul style="list-style-type: none"><li>○ Netejar esdeveniments repeits</li><li>○ Revisar classificació</li><li>○ Balanceig d'esdeveniments</li></ul></li><li>● [Fix] Arreglar xat d'usuaris</li><li>● Es llisten tots els esdeveniments<ul style="list-style-type: none"><li>○ Crear pàgina esdeveniment</li><li>○ Crear element esdeveniment</li><li>○ Crear la llista esdeveniments</li><li>○ Crer el filtre d'esdeveniments per categoria</li><li>○ Crear filtre esdeveniments per dia</li></ul></li><li>● Afegir notificacions bàsiques a usuari</li><li>● Crear diferents apartats d'esdeveniments<ul style="list-style-type: none"><li>○ Recomanants</li><li>○ Top 20</li><li>○ Random</li><li>○ Totsels esdeveniments</li></ul></li></ul>
-------------------------	---

17/10/2015 - 30/10/2015	<ul style="list-style-type: none"><li>• Crear relacions usuari-esdeveniment<ul style="list-style-type: none"><li>○ Llegit</li><li>○ Assitiré</li><li>○ M'agradaria</li><li>○ No assistiré</li><li>○ No me'l mostris més</li></ul></li><li>• Afegir polítics de cookies</li><li>• Afegir termes i condicions</li><li>• Penjar la web al AWS<ul style="list-style-type: none"><li>○ Contractar ES3</li><li>○ Contractar base de dades</li><li>○ Git clone al nou servidor</li><li>○ penjar esquema al ES3</li><li>○ Comprovar funcionament</li><li>○ Configurar seguretat HTTP</li><li>○ Configurar DNS boonobo.es</li></ul></li><li>• [Fix bugs] Errors trobats per usuaris<ul style="list-style-type: none"><li>○ No mostra tots els assistents de l'esdeveniment, només 1</li><li>○ Afegir un mini-link als esdeveniments del perfil</li><li>○ No es veu la foto de l'esdeveniment</li><li>○ Sobresurt la descripció de l'esdeveniment</li><li>○ No es guarda bé l'assistència</li><li>○ "conversa amb" del xat no mostra la imatge</li><li>○ Quan canvies perfil i guardes, la url canvia i no et permet fer més canvis</li><li>○ Error al console de javascript per HTML del xat</li><li>○ No es veu la foto de l'usuari al comentari de l'esdeveniment</li><li>○ Events bigints no acceptats per jquery (error en la redirecció)</li></ul></li></ul>
-------------------------	--

9/11/2015 - 22/17/2015	<ul style="list-style-type: none"> <li>• Entorn de desenvolupament del SR <ul style="list-style-type: none"> <li>○ Configurar Eclipse</li> <li>○ Configurar màquina virtual de java</li> <li>○ Configurar Rfor java</li> <li>○ integrar rJava</li> <li>○ Configurar Apache Maths</li> </ul> </li> <li>• Desenvolupament SR collaborative - prèvi <ul style="list-style-type: none"> <li>○ Configuració jbdc</li> <li>○ Llegir dades de la BdD</li> <li>○ Fer esbóç del programa</li> <li>○ Passar pseudocodi</li> <li>○ Testeig d'escriptura a BdD</li> </ul> </li> </ul>
23/11/2015 - 6/12/2015	<ul style="list-style-type: none"> <li>• Desenvolupament nucli SR <ul style="list-style-type: none"> <li>○ resultat d'un usuari per un esdeveniment</li> <li>○ resultats de tots els usuaris per un esdeveniment</li> <li>○ resultats de tots els usuaris per tots els esdeveniments</li> <li>○ especificar veïnat: tots usuaris o amistats</li> <li>○ Determinar un rang de presa de decisions</li> <li>○ Testeig amb data set real</li> </ul> </li> </ul>

<p>9/12/2015 - 22/12/2015</p>	<ul style="list-style-type: none"><li>• Inserir resultats finals a la base de dades tenint en compte possibles duplicitats</li><li>• Afegir filtre per contingut al SR<ul style="list-style-type: none"><li>◦ Afegir botó “no me'l recomanis més” a la vista</li><li>◦ Guardar score -1 a l'esdeveniment seleccionat</li><li>◦ Recollir preferències d'usuaris</li><li>◦ Aplicar preferències com a filtre al SR</li><li>◦ Visualitzar sortida final</li><li>◦ Testeig amb data-set real</li></ul></li><li>• Agafar un perfil d'usuari i verificar les seves recomanacions</li><li>• Recomanador en inici fred (&lt; 5 esdeveniments assitits)</li><li>• Refactorització del codi del SR</li></ul>
-------------------------------	--